Carnegie Mellon
Software Engineering Institute

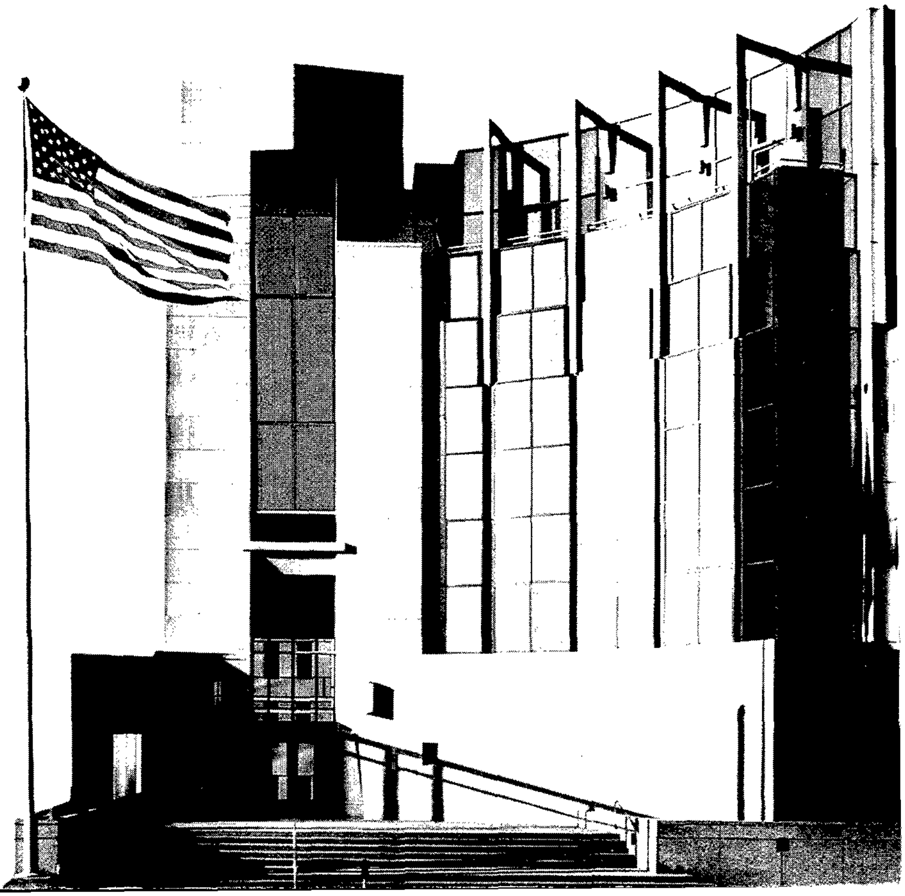
Including Interoperability in the Acquisition Process

B. Craig Meyers
Ira A. Monarch
Linda Levine
James D. Smith

March 2005

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

TECHNICAL REPORT
CMU/SEI-2005-TR-004
ESC-TR-2005-004



Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

Including Interoperability in the Acquisition Process

CMU/SEI-2005-TR-004
ESC-TR-2005-004

B. Craig Meyers
Ira A. Monarch
Linda Levine
James D. Smith

March 2005

Integration of Software-Intensive Systems Initiative

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Christos Scondras
Chief of Programs, XPK

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

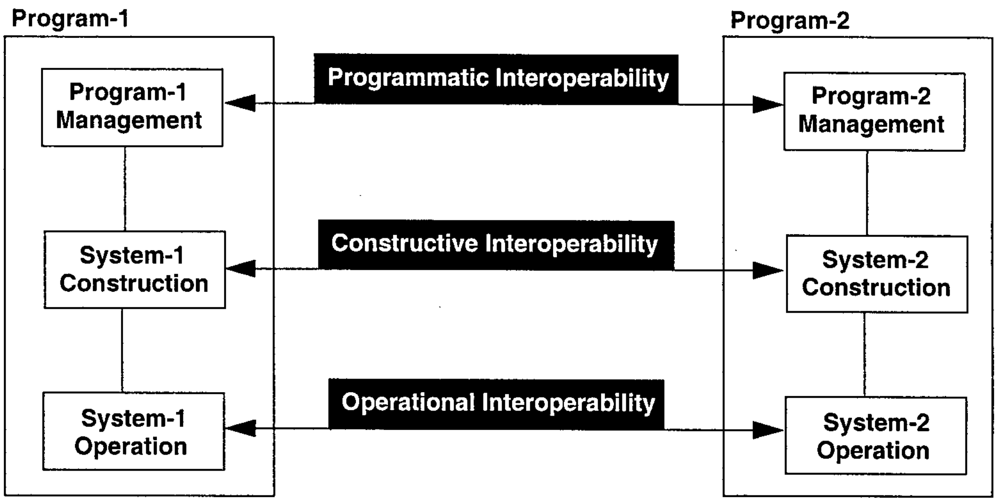
Table of Contents

List of Figures

List of Tables

Executive Summary

Interoperability has been traditionally viewed in an operational context. We believe that interoperability must also address program management and system construction. This leads to consideration of *programmatic interoperability* and *constructive interoperability.* We seek a broader scope of interoperability, as illustrated in the following figure:



This report puts forth a number of assertions relevant to achieving interoperability in the acquisition process. These include

- An *ontology* for acquisition of software-intensive systems would provide the means to specify concepts, their structure, knowledge content, and reasoning properties for multiple levels of discourse.
- An *acquisition framework*, derived from the ontology, can provide necessary knowledge applicable to acquisition.
- An *acquisition model*, derived from the acquisition framework, can be used to describe a particular acquisition project.
- An *acquisition library*, based on the acquisition framework, may be used in a project-centric context, facilitating reuse of acquisition knowledge.
- *Integration* of multiple acquisition projects can be specified using the language of the acquisition framework.

- *Experiential knowledge* can be captured in the context of the acquisition framework, fostering an acquisition learning environment.
- *Formalism* can provide a disciplined approach to reason about an acquisition.

Acquisition could benefit from a more disciplined approach. We expend major effort on the specification, development, and operation of computer systems, and see the fruits of our labor in the systems we have created. We suggest that similar rewards would be found with the acquisition system if the same skills and approaches were employed as with operational systems.

Acknowledgements

We gratefully acknowledge support of Lorraine Wilson of the Naval Sea Systems Command. We also acknowledge discussions with SEI colleagues Patricia Oberndorf, Lisa Brownsword, and Peter Capell for their useful suggestions.

Abstract

This report explores achieving interoperability in the acquisition process. It asserts that interoperability applies to the management and construction of a system, as well as to its operation. This idea leads to a broader view of interoperability. Also presented is the idea that the essential character of interoperability—related to data models and operational semantics—is independent of a domain of application. This report lists a number of basic assertions that can help organizations achieve interoperability in the acquisition process. A number of related key issues are also examined. Ultimately, it is expected that achieving interoperability will depend on a formal specification of acquisition.

1 Introduction

The activities related to the creation of software-intensive systems can be largely grouped into two categories. One aspect involves the use of technology in the construction of a system. A second aspect relates to the management practices that are employed. In the past, technology has benefitted from the use of models and formalism. The management side has not taken such an approach. We seek to remedy this dichotomy in perspective.

1.1 Intended Audience

This report sets forth a proposed approach that can lead to greater interoperability in the acquisition community. It is intended for persons interested in research in the acquisition process.

1.2 Expanding the Scope of Interoperability

Traditionally, interoperability has been viewed as a property of an *operational* system. Although we understand this perspective, we suggest it is insufficient to optimally achieve interoperability in an operational context. We believe that interoperability must also address—and resolve—issues related to the management and construction of systems, not just their operation. This is especially true for acquisition in a system-of-systems context. We seek a broader view of interoperability.

While a number of definitions of interoperability emphasize some manner of "working together" in an operational context, we submit the following:

> **interoperability:** The ability of a set of communicating entities to (i) exchange specified state data, and (ii) operate on that state data according to specified, agreed-upon operational semantics.[1]

Notice that the above definition is neutral with respect to the domain of application. It may apply to management domains, constructive domains, or operational domains.

1. *Operational semantics* refers, loosely, to the semantics of operations that are performed by an abstract machine capable of executing a specification. Operations may be defined in terms of pre- and post-conditions, whose application may result in a change of state. The meaning of the (abstract) specification then, is defined in terms of the operations that may be performed.

In fact, the focus of a recent Independent Research and Development project at the SEI was to understand the role of interoperability in a larger context [Levine 03, Morris 04]. A key aspect of this work was to develop a model of constituents that participate in the development of interoperable systems. Consider the case where there are two programs creating systems that are expected to interoperate. The different types of interoperability are shown in Figure 1.

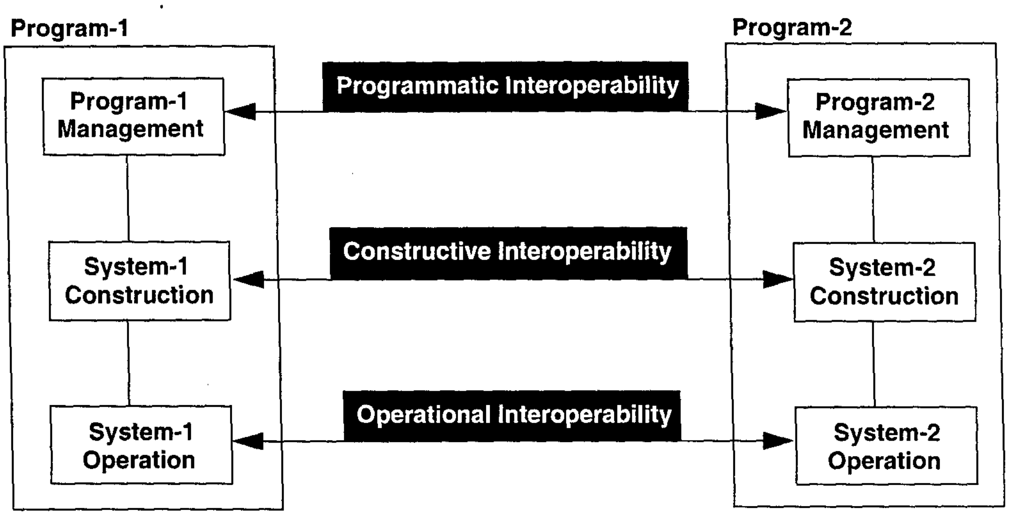


Figure 1: Different Types of Interoperability

In Figure 1 we have introduced three different types of interoperability. Our contention is that operational interoperability is more likely to be achieved if the interoperability of management and constructive aspects of acquisition are also addressed.

1.3 Formalizing Acquisition to Understand and Manage Interoperability

The acquisition of software-intensive systems is fraught with difficulty. Dealing with a system of systems brings an even greater challenge. A contributing factor can be traced to the acquisition process. While there have been improvements in the acquisition process, we believe its foundation is not sufficient to achieve a broader scope of interoperability.

Gaining deeper understanding through the use of formalism can significantly improve the acquisition environment. This may, in turn, lead to greater interoperability in operational systems. This report presents a number of assertions. We believe that by adhering to them, one may achieve interoperability in the acquisition process. The assertions are briefly stated as follows:

- An *ontology* for acquisition of software-intensive systems would provide the means to specify concepts, their structure, knowledge content, and reasoning properties for multiple levels of discourse.
- An *acquisition framework*, derived from the ontology, can provide necessary knowledge applicable to acquisition.
- An *acquisition model*, derived from the acquisition framework, can be used to describe a particular acquisition project.
- An *acquisition library*, based on the acquisition framework, may be used in a project-centric context, facilitating reuse of acquisition knowledge.
- *Integration* of multiple acquisition projects can be specified using the language of the acquisition framework.
- *Experiential knowledge* can be captured in the context of the acquisition framework, fostering an acquisition learning environment.
- *Formalism* can provide a disciplined approach to reasoning about acquisition.

The preceding assertions form the foundation for achieving interoperability in the acquisition process.

1.4 Sample Problems

To set the stage we introduce a number of problems that have occurred that help motivate this work. These problems occurred when there was an attempt to integrate multiple systems to form a system of systems.

- *Risk management:* Members of a number of programs were discussing the status of their identified risks. Some programs referred to their risk status by a color scheme, while other programs used numerical values. Others used a scheme that included the values "high, low, and medium." There was a clear lack of common vocabulary for the discussion of risk status.
- *Requirements management.* Multiple systems had been developed to their own set of requirements as well as some system-of-system requirements. During integration problems were identified. The problems were traced to the fact that there were underlying conflicts in requirements management, causing problems in integration. There was no process for identification and resolution of conflicts among requirements.
- *Reusable code*: A system was developed by reusing a lot of code from other systems. During integration it was discovered that reused code from some systems had more defects than that from other systems, causing problems in integration. There was no specification of criteria that code had to satisfy in order for it to be reused.
- *Cost:* When a number of subsystems were being integrated, problems were encountered. The programs owning the subsystems involved refused to accept responsibility and felt the

other programs' subsystems were at fault. None of the programs had seriously addressed the estimated cost of large-scale integration, not to mention which program should make the change, and who should pay.

Each of the above cases reflects some aspect of the acquisition process. In each of the examples above, the perspective of a program-centric view appears. The acquisition of the larger system suffered because of this program-centric perspective. Quite naturally, interoperability in the operational systems also suffered.

1.5 Value of this Work

It is relevant to address an important question that will no doubt arise: What is the value of this approach, based on the above assertions? Our goal is not simply to develop an ontology for the acquisition of software-intensive systems. Nor is it just to create frameworks and models or to simply apply formal mathematics to a specification of acquisition. We believe the values come as a result of the increased understanding, especially *shared* understanding, and the ability to reason about acquisition that our approach enables.

To go one step further, we believe such an approach is necessary for achieving interoperable acquisition.[2] Moreover, as a result of the ability to reason about acquisition comes the possibility of automating some of this reasoning and some aspects of the transactions enabled by this reasoning. This includes the use of autonomous software entities that are active participants in the overall process. However, the focus of this report is on an exposition of concepts, which we will address first.

1.6 Organization and Acknowledgements

This report is organized in the following manner: The assertions are discussed in Section 2, while issues are identified in Section 3. In Section 4 we explore some usage scenarios to illustrate the tenets of this work. A brief summary is found in Section 5.

2. We use the phrase *interoperable acquisition* as shorthand for interoperability in the acquisition process.

2 Discussion

In this section we provide an expanded discussion of the concepts, stated as assertions, relevant to achieving interoperability in the acquisition process. Some discussion of the integration of the concepts is also provided to give a higher level perspective.

2.1 Ontologies

> *There would be significant value in the development of ontologies for acquisition of software-intensive systems. Ontologies provide a means to specify concepts, their structure, knowledge content and reasoning properties for multiple levels of discourse. A formal ontology would provide needed conceptual resources for specifying an acquisition framework and deriving acquisition models with the aim of establishing data models and operational semantics for support of programmatic, constructive, and operational interoperability.*

The study of ontologies has been around since the work of Aristotle and the Greek philosophers. Ontologies are developed for many domains, such as medicine, libraries, or law. There has also been a significant amount of work in ontologies for information and software-intensive systems. Ontologies are also considered for application to Web services.

Since much of this report will relate to ontologies (in this section, as well as acquisition frameworks and the specification of acquisition practices) it is worth spending some time on a general description of ontologies. Essentially, an ontology is a way to organize and reason about knowledge in some domain.[3] We offer the following definition:

An *ontology* is a form of knowledge representation that includes specification and organization of

- concepts
- structural relations among concepts, typically expressed as a taxonomy or semantic network

3. Certain terms are often found in discussion of ontologies. A *foundation* (or upper) ontology refers to a higher level, or cross-domain, perspective. The term *domain ontology* is used to refer to the development of some ontology to a particular domain. When a domain ontology is combined with a foundation ontology, it is often referred to as a *core ontology*.

- concept metadata (e.g., data that can be expressed in frames along with other elements of the representation)
- reasoning properties, expressed at multiple levels
  - ontology
  - concept data

Concepts

A fundamental aspect of an ontology is expressed by concepts or types and instances. We include instances of types in models. Ontologies are related to models as types are to instances. These concepts form the domain of discourse with which the ontology is concerned. For acquisition, we may include the concept of a contract, an award fee, or a system engineering master plan.

Concepts are sometimes called *universals* in the philosophy literature. Instances are *particulars* of those concepts. The relation here is reminiscent of an object as an instance of a class in an object-oriented approach.[4] For example, a specific contract is an example (or particular) of the concept of a contract.

Relations Among Concepts

Another element of an ontology is manifest in its structure, frequently presented as a taxonomy. For example, the work within the *DOLCE* (*Descriptive Ontology for Linguistic and Cognitive Engineering*) community, is organized in a taxonomy whose structure appears in Figure 2 [Masolo 03]. The term endurant refers to things that are enduring, such as a physical object. On the other hand, perdurant refers to things that have a characteristic temporal behavior, such as a process, which can have different states at different times.Only a part of the ontology is shown here.[5]

Relations among those concepts are an important aspect of an ontology. For example, one may frequently see the relations *part_of* or *used_by.* An engine is *part_of* a car.

Relations among concepts may be presented as a semantic network, as illustrated in Figure 3. We show a small example of an activity related to software development. Note that software development *is_a* activity, indicating a relation between software development and an activity.

---

4. Note that in the discussion of ontologies, there is considerable richness in the relation between a general (or universal) concept and particular concepts. We will not go into the details here; suffice it to say, however, that much of the literature dealing with ontologies contains an expressiveness that is not often found in the object-oriented approach. Of course, ontologies have a tad of a head start!

5. We do not necessarily commit to the use of *DOLCE* for the interoperability work for various technical reasons, but simply cite it as an example of an ontology.

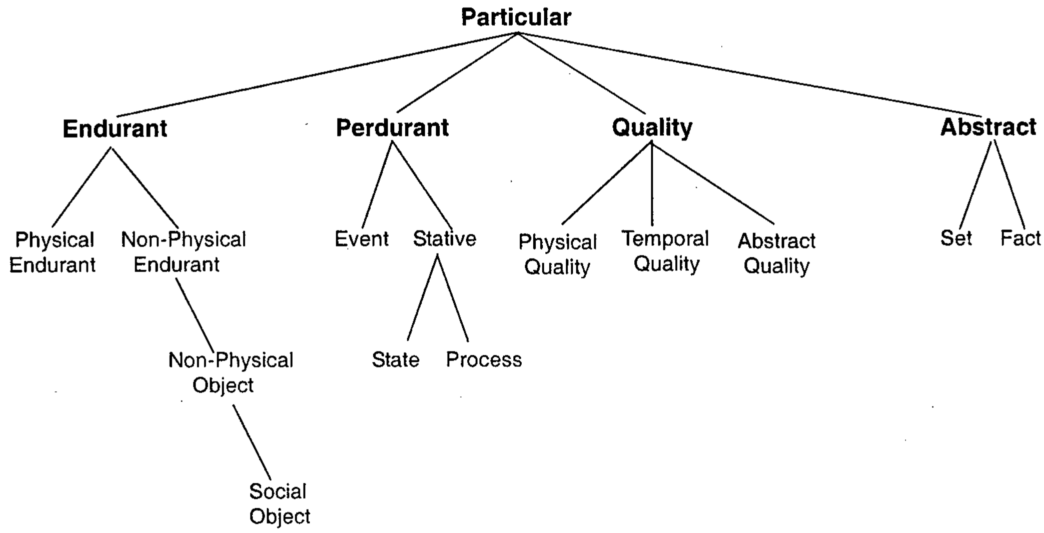


Figure 2: Illustration of DOLCE Taxonomy

Note also that Joe Smith *is_a* person *responsible_for* software development. The purpose here is to develop and understand relations between concepts, both universal and particular.

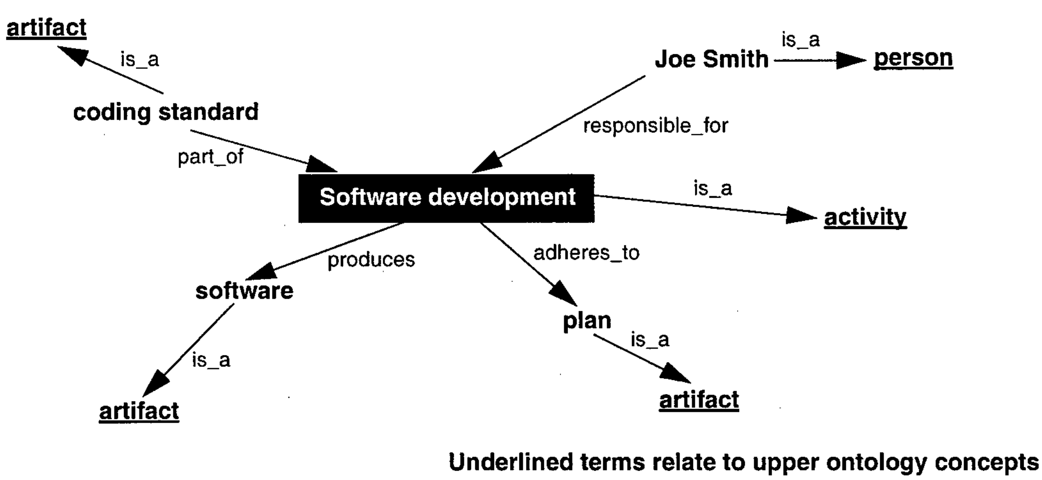


Figure 3: Illustrating Relations Among Acquisition Concepts

Concept Metadata

Another aspect of an ontology is the expression of content for a concept. Several approaches are possible; one is based on the notion of *frames*. A frame serves to encapsulate information about either a concept or a particular in the ontological structure. A frame has associated *slots*

which may assume values. The slots of a frame represent attributes or relations. In computer science, artificial intelligence in particular, frames were used for knowledge representation in automated reasoning systems.

There are languages that provide for the development of frame-based systems. For example, in *FrameNet*[6] an *activity* includes some of the frames shown in Figure 4. Notice that the specification of the activity includes reference to other frames, such as one for *Activity_Start*.

Activity

Definition: This is an abstract frame for durative activities, in which an Agent performs an intentional Activity: entering an ongoing state of an Activity, remaining in this state for some Duration of Time, and leaving this state either by finishing or stopping. It is used mostly for inheritance of common framework elements, and provides the frame structure for beginning, ongoing, finish, or stop of intentional activities, each of which constitutes a subframe of this frame.

Inherits from: Process
Is Inherited by:
Subframe of:
Has Subframes: Activity_done_state, Activity_finish, Activity_initial_state, Activity_ongoing, Activity_pause, Activity_paused_state, Activity_prepare, Activity_start, Activity_stop, Activity_unfinished_state
Uses:
Is Used By:
See also:

Figure 4: Sample Frame for an Activity from FrameNet

The use of frames to represent content can be related to our earlier example shown in Figure 3. There, we included "Joe Smith" as being responsible for software development. Since he is a person, he would have characteristics associated with a person. That is, there would be information about a person, organized in frames. It might include the name, email address, and cell phone number. In this case, "Joe Smith" would represent a particular person.

Although the term *frame* has taken on a specific intent, we prefer to view it as a general means to encapsulate information about some data associated with a concept. There are clearly vari-

6. FrameNet also finds application in the study of linguistics. See, for example, http://www.icsi.berkeley.edu/~framenet.

ous types of information that one can envision for some data associated with a concept, and the term frame, as typically employed in the general literature, may be too limiting.[7]

Reasoning Properties

The final element of an ontology deals with the specification of reasoning properties. Reasoning properties can appear in two ways.

First, there may be a need to reason about the ontology as a whole. For example, we may wish to state that all activities within the scope of some project are adhering to the project software development plan. Or we may desire to reason about the state of some collection of artifacts that are used in some system, perhaps including those provided by commercial vendors. In each case, we require a formalism that is expressive enough to deal with the scope of problems we seek to address.

The second case is where one seeks to reason about the characteristics of some concept. These can be expressed as axioms sometimes embedded in a frame. For example, if a concept referred to a starting and stopping time, it may be desirable to state that the starting time must be before the stopping time. This may be viewed as reasoning in a local, concept-specific context. These reasoning properties are typically specified in a predicate calculus.

A variety of approaches have been used to address reasoning properties. These range from natural language to the use of a formal specification. We prefer the latter as it provides a more consistent and concise approach.

Summary

The basic elements of an ontology, namely, concepts, structure, the expression of knowledge content, and the approach to reasoning, all apply to the development of ontologies for acquisition. Note the relation here to the essential characteristics of interoperability: specification of data and operations performed on that data. We are coming full circle.

7. In some unpublished work we have explored the use of types of frames, where the type is defined by its purpose and information context. For example, we have explored the use of expository frames (for general information about some concept), declarative frames (for information about state data associated with a concept in terms of its properties), axiomatic frames (which are logic statements over the content specified by a declarative frame), experiential frames (containing experiential knowledge), and configuration frames (which contain information such as the author, date, and version of the frame data). The degree to which these various types of frames can apply to this work requires further consideration.

2.2 Acquisition Framework

An acquisition framework can be defined on the basis of an (upper) ontology. The framework would provide support for specifying acquisition concepts, such as activities, internal and external events, as well as entrance and exit criteria. Although concepts of the acquisition framework have a temporal aspect, the framework does not have any built-in preference for a life cycle model, as that is a local matter.

The discussion of the acquisition framework involves two related ideas. First, there is a domain ontology for acquisition. Then, there is an identification of elements of that domain, which provides the acquisition framework. Since it deals with acquisition-specific concepts, we include domain ontology here, as opposed to in the preceding section,

Domain Ontology

As noted earlier, a domain ontology is an ontology that is specific to some domain, in our case acquisition. A domain ontology is created from an (upper) ontology by identification and further delineation of those aspects related to a particular domain. As we will see, a domain ontology maintains some generality, while the framework provides more information relative to the domain.

In previous work we have developed an acquisition framework, although that should more properly be viewed as a domain ontology [Meyers 01a]. We applied those ideas to the development of an acquisition model for standards-based, COTS-products [Meyers 01b]. Our main goal in that work was to be able to understand and reason about acquisition, the same way one desires to understand and reason about an operational system. For example, we often hear of a waterfall model [Royce 70] or a spiral model [Boehm 88], or an evolutionary model. Yet can one precisely define what these terms mean, and what their differences are? How can one, *formally*, reason about these different acquisition approaches?

Our work on the acquisition framework introduced the following concepts [Meyers 01a]:

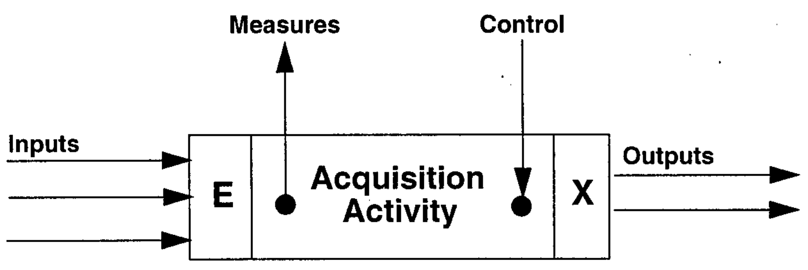
- activities, including their relation to other activities
- internal events (i.e., those that are initiated within the scope of the project, such as a risk review)
- external events (i.e., those that are initiated outside the scope of the project, such as a COTS product upgrade)
- requirements
- system instances (e.g., a build of a product)

The preceding concepts were required in order to minimally specify a model for acquisition.[8] Additionally, we provided optional concepts related to participants and artifacts, as well as entrance and exit criteria. A summary of the acquisition concepts appears in Table 1.

Table 1: Summary of Acquisition Elements

Required Elements	Optional Elements
Activities	Artifacts
External Events	Entrance Criteria
Internal Events	Exit Criteria
Requirements	Participants
System Instances	

The concept of an acquisition activity is shown in Figure 5. We simply show an activity that accepts some inputs and produces some outputs. Entrance and exit criteria are also associated with the activity.



Note: E and X denote entrance and exit criteria, respectively.

Figure 5: Simple Model of a Process

Also shown in Figure 5 are *measures* and *control*. The measures are representative of the execution of the activity itself (and typically not treated as an output used by some other activity). The inclusion of control allows for operations to be performed on the process, such as starting or stopping the process.[9]

8. The reader may wonder about the choices of acquisition elements. However, recalling that our goal was to specify the information necessary to construct an acquisition model may help to clarify things. For example, key elements that differentiate a waterfall model from a spiral model are the mapping of activities to time, as well as the mapping of requirements to instances of a system.

Because the emphasis is on higher level concepts, we did not identify any particular activity, such as program management, or software development. Further, we did not explicitly account for any life-cycle considerations. However, various *models* could be constructed by mapping acquisition concepts onto time. Thus, one could specify a sequence of activities that may be representative of a waterfall model. Similarly, a spiral model or evolutionary model could be developed. However, the choice of any particular life-cycle view is a local decision.

We also included a specification of operations that may be performed on the acquisition concepts. For example, an operation is provided that can initiate an activity, or declare that some requirement is satisfied. Notice that the treatment in terms of data elements and operations on that data is purposefully consistent with our general view of interoperability.

Meyers also discussed a specification of criteria that may be applied to assess a particular acquisition [Meyers 01a]. Such a specification was based on a formal approach. We defined, for example, what it means for an acquisition model to be *complete* (for example, every declared internal event must be handled by some activity). This point is further illustrated in Section 2.7.

Framework Elements

The elements of the acquisition framework are obtained from the domain ontology. A framework element bears a *kind_of* relation to a corresponding element in the domain ontology.

For example, the domain ontology contains an acquisition activity. The framework needs to identify the kinds of acquisition activities. Some candidate activities might include

- risk management
- requirements management
- contract management

A similar remark may be made about other elements in the domain ontology. Some candidate internal events might include

- budget review
- risk review
- critical design review

9. Inclusion of a control (interface) brings an executable character to the process. This further implies the existence of a management agent that is capable of initiating operations on the process. Note that the process itself is intentional—it does not run by itself; i.e., there are operations *in* the process not just *on* it. We now tend toward a view of executable acquisition with the interesting question of achieving interoperability in the acquisition process.

Also, some candidate artifacts might include

- system engineering master plan
- system test plan
- reuse management plan
- risk management plan

As noted above, the elements of the domain ontology (activities, events, etc.) may have attributes. The specification of attributes for a concepts can be provided through the use of frames as discussed on page 7. They remain incompletely specified until a particular model is created.

2.3 Acquisition Models

It is possible to specify an acquisition model which is derived from the acquisition framework. An acquisition framework is general, but an acquisition model is project specific. An acquisition model inherits the properties of the framework. Conformance of an acquisition model to the acquisition framework is especially important.

The key points here are that a model is derived from the framework, and that it is expected to conform to the framework. A concrete example of how a model relates to the framework is relevant. An element of the framework is expected to have certain attributes. In the context of the framework, these attributes can be viewed as *slots* (recall the discussion of expression of knowledge content through the use of frames in Section 2.1) that are incompletely specified. However, from the project view, the attributes would need to be specified. For example, Table 2 shows the application of this idea for an event. Note that this particular type of event would be an *external* event since it is initiated outside the scope of the acquisition project.

Table 2: Sample Event Data for Acquisition Framework and Model

Acquisition Framework	Acquisition Model
Name: ___________	COTS OS Upgrade
Source: ___________	Company XYZ
Date of First Occurrence: ___________	10/1/2004
Date of Last Occurrence: ___________	TBD
Frequency: ___________	Quarterly
Related Activities: ___________	COTS Product Acceptance Testing, COTS Product Licensing
Project POC: ___________	Joe E. Acquisition

The apparent template completion exercise shown in Table 2 appears almost trivial. But there is a deeper significance than one might first think. We'd like to make several points:

- Simply recognizing and accounting for the existence of the event—for example, a new version of a COTS operating system—is a first step of some importance. In essence, the project has accepted responsibility for managing the event.
- Knowledge of the temporal interval over which the event can occur is important for planning purposes: How will the project respond to the external event? What resources will be allocated? A similar remark might be made about the estimated frequency of occurrence.
- Identification of the related activities is important. This tells us the activities that are expected to be performed in response to the event. So if a new version of the COTS operating system is provided, we indicate that the (project-defined) activities related to acceptance testing and COTS product licensing will be invoked. These activities are part of the acquisition effort to deal with the external event. Presumably, there are other activities that follow from these; for example, if the upgrade passes acceptance testing, it may be given to those responsible for system integration.
- Other relevant information is captured here, such as the points of contact for the (external) organization initiating the external event, as well as someone within the project who is responsible for dealing with the event.

Hence, although a model created from the framework has a seemingly simple nature to it, there is more than meets the eye. An additional consideration comes from the fact that the model must conform to the framework. This means that the requirements and checks provided in the framework apply to a model as well; we'll discuss this a bit more in Section 2.7.

We may illustrate the temporal nature of an acquisition model by consideration of the waterfall process. Of course, since the framework is silent with respect to life-cycle considerations, any other model could equally be used. The primary characteristic of a waterfall model is that it is traditionally viewed as a series of sequential activities. To this we will add a continuous activity whose function is management of the other activities. Figure 6 shows a simple diagram.

For simplicity we do not show the inputs and outputs of the activities within the process, nor do we show the measures and control aspects. Further, the only relation we have indicated is that between the overall management activity and the other activities. Each activity dutifully goes about its business, interacting only with the management process. Life is good.

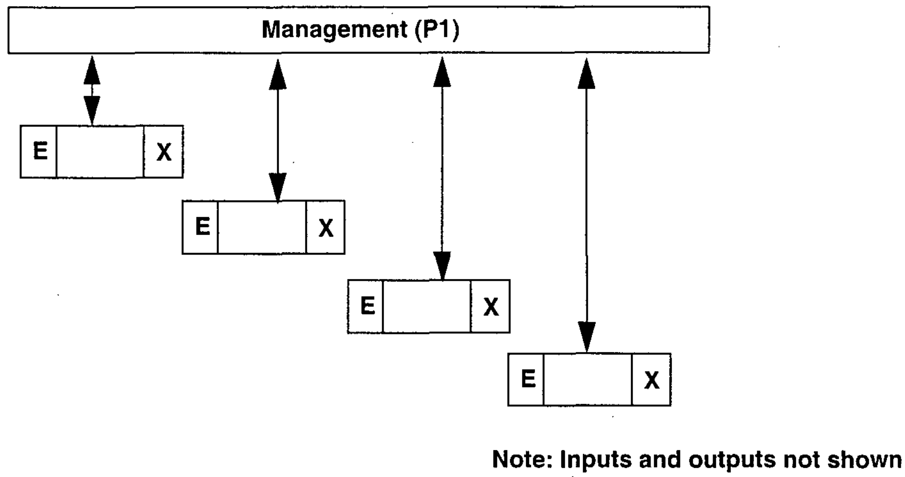


Figure 6: Simple Example of a Waterfall Model

2.4 Acquisition Library

A library of acquisition practices would help to further the goals of interoperability in the acquisition process. Such practices would be cast in the form of the acquisition framework. An acquisition practice may also be viewed in terms of a domain-specific ontology, derived from the upper ontology. This information may then be incorporated into a project acquisition model, providing facilitated reuse of acquisition knowledge.

We have all heard of best practices relating to various aspects of acquisition, and much effort is devoted to the identification and development of such practices. There has been a trend in recent years to use practices that are based on industry standards. For example, there is an IEEE standard for risk management [IEEE 01]. Another source for community-based information is that provided in the Capability Maturity Model® Integration (CMMI®)[Chrissis 03]. It is evident that there is much material on which to base a common practice for aspects of acquisition.

A key notion in our approach is that the specification of some best practice should be developed in the context of the acquisition framework. We take this approach so that the model developed from the best practice can be more easily integrated in the context of the acquisition framework.[10]

Consider the case of a best practice, or standard, for risk management. From the perspective of the domain ontology, we would need to answer questions such as

- What are the activities? Typical activities might include risk identification, risk assessment, and risk planning, among others.
- What are the temporal characteristics of the activities? For example, when do activities start and stop; this takes on a life-cycle perspective.
- What are the relevant artifacts? For example, a risk management plan is frequently developed.
- What are the external events? One candidate here might be an event related to dealing with risks outside the scope of the project, but which might affect the project, such as existence of a new COTS product.
- What are the internal events? For example, a risk review is a likely candidate.

The specific activities identified constitute the kinds of activities that will appear in the acquisition framework. Other aspects of the acquisition framework, described in Section 2.2, such as events and so on, would need to be addressed in the translation of a best practice to an acquisition framework.

Several important issues loom on the horizon. First, we do not expect a best practice to be complete with respect to the amount of information it provides. For example, an activity in the acquisition framework has a start and end time. Assuming that such times are recognized in the best practice, we would not expect a particular value to be specified. Such information would need to be completed by a project team. Recall the discussion of Table 2 on page 13.

The second issue deals with choice and representation of data related to the best practice. For example, in the case of a risk management practice, it may state that a risk should (or shall?) have an associated status. However, the best practice may be silent with respect to the particular values of risk status. Frequently, a color scheme of "red, yellow, or green" may indicate particular values. Of course, a scheme based on the integers one through five may also be acceptable; again the best practice may be silent with respect to a *particular* choice.

But now here comes the problem. If we are to have interoperability between acquisition projects with respect to risk *status*, how much detail must be specified? We view this as a question of compatibility, and it relates to potential difficulties in interoperability. If one project uses a "red, yellow, green" scheme, can it interoperate with another project that uses a scheme of values one though five?

---

10. Another reason for the importance of the acquisition practice: presumably the *semantics* of a practice would be known, and hence more likely to be integrated with other practices.

® Capability Maturity Model and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

We also assume that an acquisition practice may be included by a project as it develops its project-specific acquisition model. We expect to gain reuse of acquisition knowledge by this approach. Notice that a specification of an acquisition model actually exists on two different levels. On the one hand, the model is defined by the concepts used for its specification, whether those concepts are developed by the user or included via some practice. At this point, the model is still general. But as a project team begins populating the general model it begins to develop the details of its specific model. Another perspective is to view these as incomplete and complete specifications.

The point here is not just to find best practices. A more important point deals with the specification of those best practices in terms of the acquisition framework. Finally, the question of how the relevant data is chosen is also important. Information is progressively added and we move toward a particular acquisition model for some project. All of the preceding discussion is simply a reflection of a greater issue: Interoperability is a different, and more difficult, problem than just using a standard.[11]

2.5 Integration Process

The integration of multiple acquisition projects is a necessary condition for successful interoperability in the acquisition process. The integration process can be specified in terms of the concepts provided by the acquisition framework. This provides a consistent, unified approach to integration of acquisitions such that interoperability is possible. The integration process also provides specific concepts through which to evaluate and compare different acquisition models.

The need to integrate various acquisition models such that interoperability is achieved introduces a host of well-known problems. Specification of the integration of multiple projects in terms of the same language used for the specification of each of the individual project-specific aspects is expected to enhance the chance for interoperability. Reuse of framework concepts used in acquisition models is expected to provide a consistent approach to integration of these models.

An example of an integration process is shown in Figure 7 for the simple case of two projects proceeding on a waterfall path. In each case, a number of technical activities are shown, as well as an activity focused on project management. The open ovals at the center of the figure suggest the processing necessary to integrate the activities, and the interaction is represented

11. Although interoperability frequently employs standards, interoperability is more than just using a standard. For example, interoperability includes activities that give meaning to a standard and enable it to be used. Interoperability also provides a theoretical framework (i.e., vision) that situates various standards in a larger perspective.

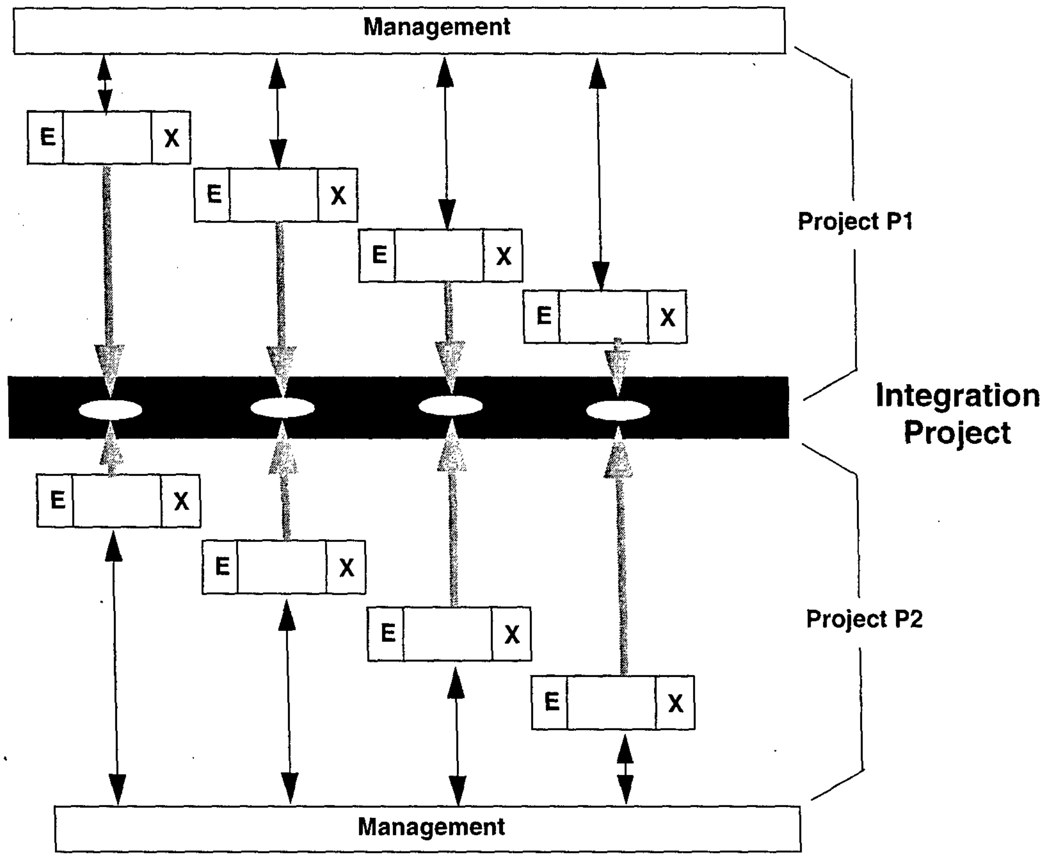


Figure 7: Integration of Waterfall Models

by shaded arrows. The case shown is for pair-wise project activity interaction but other, more complex, interactions are possible. For a pragmatic example, projects P1 and P2 may be developing software, and the integration process is responsible for the integration of the code produced by the individual projects.

We have not indicated integration of management activities for multiple projects in Figure 7. This is easily done however. If each project (management) includes an activity for risk management, then the integration might also have an activity that would integrate the project-specific risk management activities.

The assertion that the elements of the acquisition framework may be applied to specify project integration allows us to head toward interoperability among projects. We may therefore view an integration project as one whose purpose is to simply integrate other projects. A notional snippet of an integration process is shown in Figure 8.

The sketch of the integration process, at the center of Figure 8, illustrates the relation between entrance or exit criteria for the individual projects and the entrance or exit criteria in the inte-

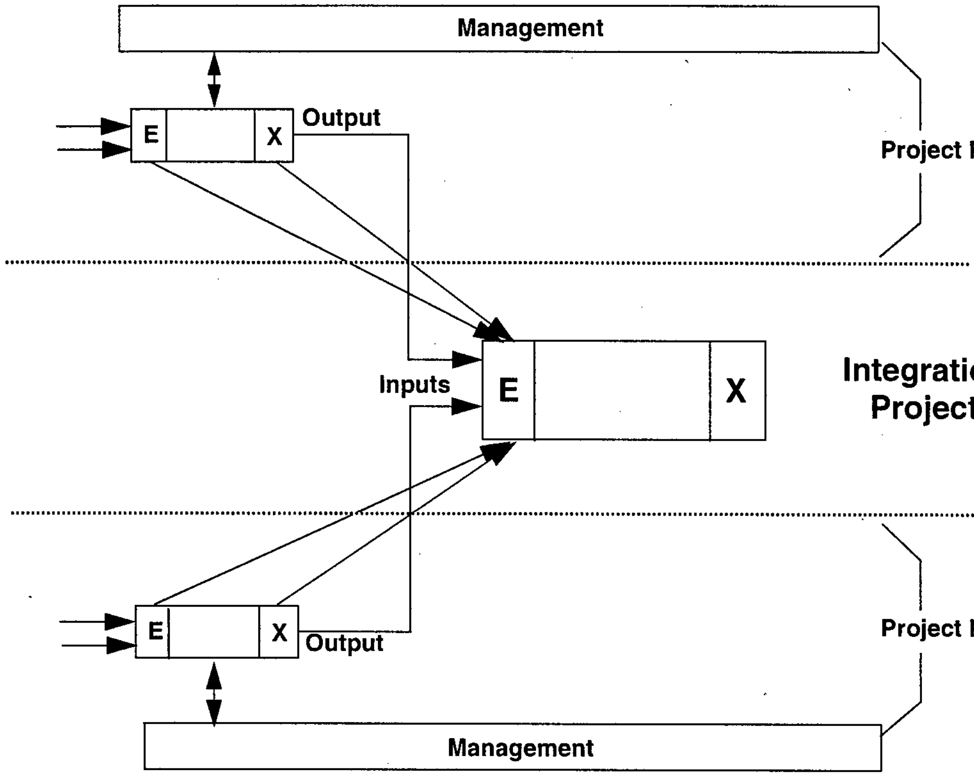


Figure 8:   A Snippet of an Integration Process

gration of those projects. For example, the entrance criteria for the integration could easily depend on the exit criteria for the projects intended to be integrated.

We emphasize that the integration of projects, shown in the center of Figure 8, is itself based on elements of the acquisition framework. That is, we have defined an activity and indicated its inputs and outputs, as well as a specification of entrance and exit criteria. This can be viewed as reuse of the acquisition framework in an integration context.[12]

It is relevant to recast the preceding discussion in the context of ontologies. We take this approach since ontologies can be used at many places in the development of the models we wish to integrate. The integration of acquisition projects can be viewed as the integration of

12. But consider a seemingly simple problem. Multiple acquisition projects have a schedule, which includes some milestones. The integration of these projects includes an integration of schedules. Now we must ask ourselves if the milestones associated with the individual projects have the same meaning. Such meaning can be defined in an ontology. Hence, the integration of schedules can be viewed, at last in part, as an integration of ontologies.

ontologies. A general discussion of integration of ontologies can be found in Wache [Wache 01].

Figure 9 shows two simple approaches to model integration. The first, shown in part (a) of the figure, is based on integration of the models through their use of different ontologies. In part (b) of the figure we show the integration through the use of a common ontology used for both models. There are strengths and weaknesses to each approach; a general discussion appears in Wache [Wache 01].

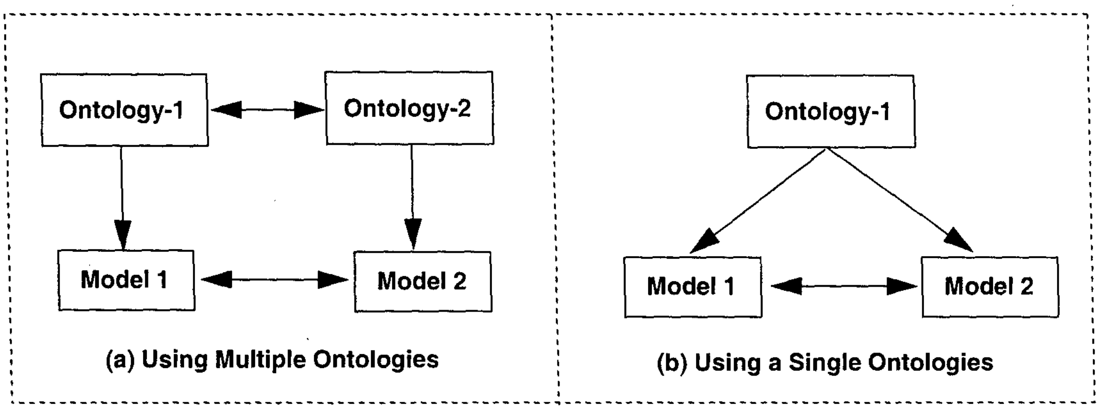


Figure 9: Approaches for Integration of Ontologies

2.6 Experiential Knowledge

It is important to consider experience of organizations engaged in acquisition. Such experience could be cast in the context of concepts of the acquisition framework, thereby providing cognitive support for organizing and accessing these experiences. The experiential knowledge would be available to others, fostering an acquisition learning environment.

The notion of an experience factory was presented by Vic Basili, [Basili 94]. The concept provides a structure within which experiences can be captured to support learning. One can also develop models of the experience base and use that information to understand and predict future developments.

While we accept the original purpose and idea of an experience factory, we want to take the idea further. First, we believe it is important that experiential knowledge be provided in a particular *context*. Second, we seek to allow the experience factory to be used by many organizations rather than single organizations (since the experiential knowledge is based on the acquisition framework). The view that emerges of an experience factory is shown in Figure 10.

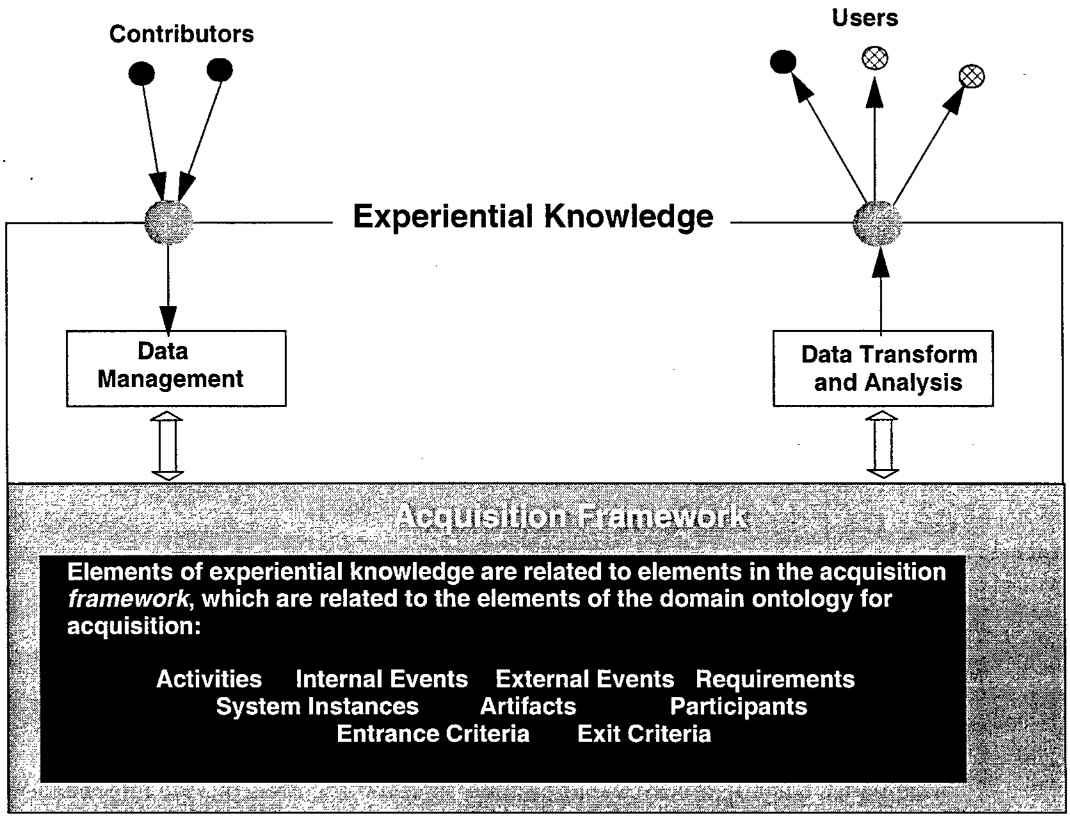


Figure 10: Notion of Experience Factory in Acquisition Context

Projects may contribute information to the experience factory. Users also interact with the experience factory as consumers of its data. Note that a contributor is in many cases a user, but the converse may not be true. The two major functions that the experience factory provides are managing data (from contributors) and performing analysis of that data (at the request of users).

Our approach emphasizes the use of an acquisition framework, shown at the bottom in Figure 10. Data may be entered to the experience factory *in a context defined by the acquisition framework*. Thus, we partition the experience factory according to the aspects defined by the acquisition framework that accommodates different model specifications.

For example, the domain ontology for acquisition includes a concept of *entrance criteria*. It is assumed that the entrance criteria is associated with some activity. In the context of an acquisition framework, there may be a kind of activity for *Software Reuse*. There may also be an entrance criteria which states that: “Any software considered as a reusable asset shall have fewer than 10 defects per thousand lines of code.” Some other project team, with experience in selection of reusable software, may choose to enter the following experiential knowledge: “Reusable software should identify the cost of fixing any outstanding defects.” The implica-

tion here is that just a knowledge of defects is incomplete and that what is also needed is the cost to bring the software up to a level of acceptability for a particular use. This illustrates how experiential knowledge can contribute to the betterment of the common good; a new entrance criteria might incorporate this knowledge in future efforts.

We have alluded to the fact that interoperability among projects may be obtained through the use of models which conform to an acquisition framework. We would also like to achieve interoperability among bodies of experiential knowledge so multiple organizations might gain common benefit. This is best achieved if the experience factory is structured such that it reflects the structure of the underlying acquisition framework. Hence, the sharing of experience is achieved through the shared use of the acquisition framework.

2.7 Role of Formalism

Formalism can provide a disciplined approach for reasoning about interoperability in the acquisition process. It is through a formal specification that one may define and reason about the behavior of an acquisition framework or model, or further, interoperability in the acquisition processes.

The use of mathematics has not appeared—until now—and for good reason. We understand the concern of the general reader to avoid mathematical descriptions! But we ask the reader to think again, and be patient.

As we explained in Section 2.2, one of our main reasons for developing an acquisition framework was to gain a better *understanding* of acquisition. We wanted a means to *specify* and *reason* about acquisition. This naturally leads to a formal (i.e., mathematical) approach. Our original specification of an acquisition framework was couched in formalism [Meyers 01a]. In addition to elements such as activities and events, we included operations on these items (which are really just data types). *But the emphasis has always been on understanding and reasoning, expressed through a formalism.*

We suggest that there are sound reasons why a formal approach brings value to this work. Traditionally, one hears of the ability to be concise and precise about some specification; we concur that this is valuable. But the formulation of the acquisition framework included additional material. For example, we are particularly interested in the concept of a *well-formed* acquisition model. Thus, the expressions below were included [Meyers 01a]:

- For every external event declared in the acquisition framework, there must be an activity that is responsible for processing the external event.
- For every internal event declared in the acquisition framework, there must be an activity that is responsible for processing the internal event.

- Every activity must be related to at least one other activity.
- No activity may end at a time before it starts (correctness criteria).

The key here is that through a formal approach we have been able to specify an acquisition framework. Since an acquisition model is based on the framework, the acquisition model inherits the formalism from the framework. We have not yet addressed the integration and interoperability among acquisition models; we view these as an important issues.

So to those who wonder at the utility of a formal approach—including in the acquisition process—rest assured that in the end, it is the formal approach that binds all of the preceding assertions together!

2.8 Integration of Concepts

The preceding material has illustrated concepts, stated as assertions, that we believe can assist in achieving interoperability in the acquisition process. The integration of these concepts is worth discussing.

We begin with the thread that includes ontology, framework, and model. This sequence is based on a *refinement* process starting with the ontology, from which is developed an acquisition model and is shown in Figure 11. The top plane of the figure represents the upper ontology (for which we have used the *DOLCE* form, as illustrated earlier in Figure 2 on page 7). The next plane represents the acquisition framework, derived from the ontology. Below this is an acquisition model for a *particular* project.

In the last plane of Figure 11 we show experiential knowledge. The structure for encoding the experiential knowledge is based on the elements of the acquisition framework. As illustrated in Figure 11, some of the experiential knowledge applies to the particular acquisition model, while some experiential knowledge does not apply.

Another point related to the integration of concepts is shown in Figure 12 in terms of a semantic network. The goal is to integrate two acquisition projects, defined by their respective acquisition models, shown in the shaded rectangle at the center of Figure 12. The major elements shown in this figure depict the following.

- Acquisition Model-1 contains an activity for Contract Management, as well as a contract, both of which are defined in accordance with the framework.
- Acquisition Model-2 contains an artifact representing a COTS product. A *kind_of* COTS product, labeled "Product ABC," is shown.

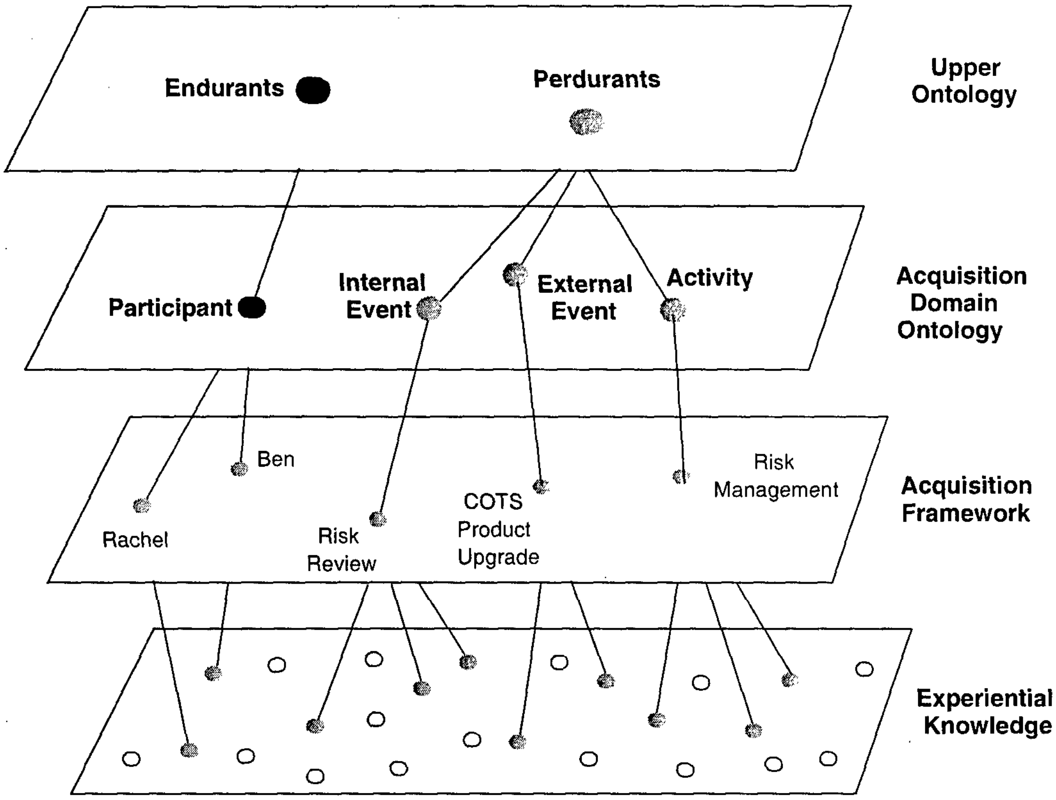


Figure 11: Summary of Integration Approach

- Acquisition Library includes a practice of risk management. Notice that the practice refers to a particular artifact of a *Risk Management Plan*, as well as a particular event of a *Risk Review.*
- Experiential knowledge contains information about a particular COTS product.

Figure 12 also shows the broader scope of interoperability in the acquisition process, as illustrated by the following statements:

- Acquisition Model-1 has a contract with an organization that *uses* the *same* COTS product that is being used in Acquisition Model-2.
- The Acquisition Library includes a practice of *Risk Management.* This practice is imported by Acquisition Model-1. Thus, for example, Acquisition Model-1 would now have an activity for *Risk Identification*, and an event for a *Risk Review.* The specification of the practice is general and would require adjustment for the context of a particular project.[13] Notice how the acquisition library can afford reuse of acquisition knowledge.

- The Experiential Knowledge includes knowledge about "Product ABC." Note that this is the same product being used in both acquisition models, indicating possible benefit of the experience related to this particular product.

Even at the simple level indicated in Figure 12, we already see a coupling among the acquisition projects. This coupling is *implicit* in that both projects are using the same COTS product. Many questions will stem from this. For example, what happens when there is an upgrade to this product? Will both projects synchronize their efforts to continue to use the same product, or will they go their separate ways? More importantly, what are the consequences of their actions to future acquisition?

Not shown in Figure 12 is any indication of how interoperability in the acquisition process can be achieved. We have asserted that the integration of multiple acquisition projects can be based on the acquisition model (derived from the framework). However, the precise specification of what it means for two acquisition projects to have a viable integrated schedule is not addressed. We believe, however, that the specification of acquisition (leading to interoperable acquisition) can be based on a formal approach, using the concepts illustrated above.

The example shown in Figure 12 is just the tip of the proverbial iceberg. A full specification of an acquisition, even for a single project, must account for many concepts, as well as attributes of those concepts. Increasing the scope to account for interoperability in the acquisition process represents a further scaling of both the problem and requirements on its solution.

2.9 Summary

In this section we've described a number of concepts, stated as assertions, relevant to achieving interoperability in the acquisition process. It is especially important to understand that the concepts are independent of a particular domain, such as project management or system construction. The premise of application domain independence is fundamental, as it allows us to apply the assertions in different application contexts. It enables a reuse of ideas.

13. For example, the acquisition practice for risk management may include activities. Two attributes of an activity are the start time and stop time. The particular values of these times would be identified by the project that imports the risk management practice from the acquisition library. It is an open question as to whether the acquisition library should contain particular values of these times. We say this because such times are naturally project specific, but knowledge of the relative times for the practice may be of value to a particular project. That is, knowing how much time was spent on some activity could be valuable to a project in its reuse of an acquisition practice.

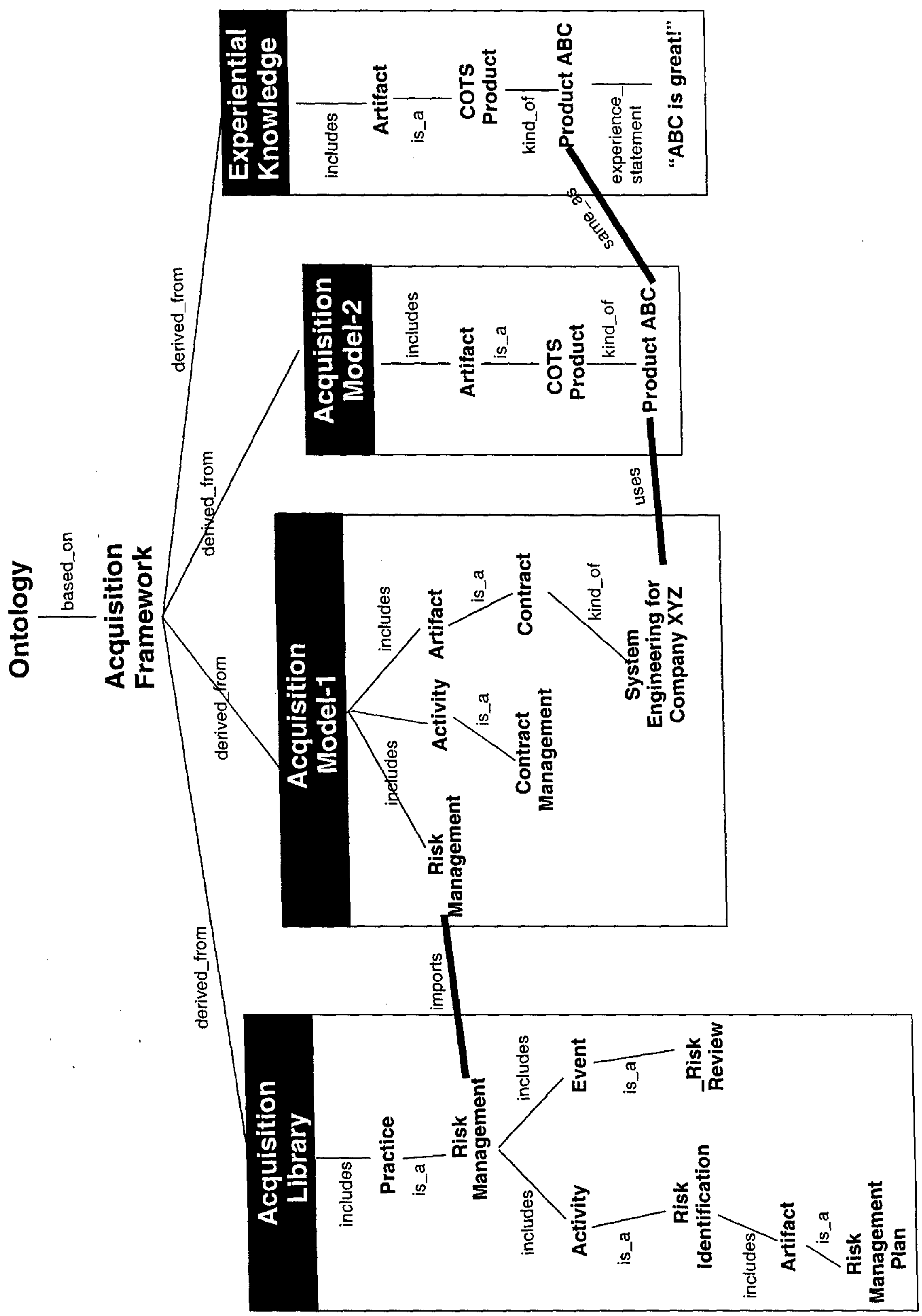


Figure 12: Assertions for Interoperability in the Acquisition Process

3 Issues

This section discusses a number of issues related to the assertions about interoperability in the acquisition process. We begin by discussing some general issues. Many of the following issues cut across different assertions; we have tried to list the assertions where most appropriate.

3.1 General

Is the current project-centric view of interoperability sufficient to deal with the larger context of acquisition? We have focused on a *project* as the entity engaged in acquisition. This leads us to consider programmatic, constructive, and operational interoperability. The approach we have taken starts with a project but cannot end there.

We have not explicitly accounted for the larger context that may require consideration. For example, if we introduce the context of upper management, how would things change? One might assume, for example, that upper management is responsible for the development of policies and procedures that apply to the projects within its purview. Can upper management be viewed simply as a "project" that interacts with other projects? We now recursively face the question of interoperability between different upper management organizations and projects that do the work to achieve acquisition.[14] These are questions of scope that bound the acquisition context.

What are the implications for interoperability within the scope of a particular project? The principal organizational element of the approach taken here is a project. When we speak of programmatic interoperability we refer to establishment of interoperability between two different projects; see Figure 1 on page 2. There are, however, interoperability concerns *within* the scope of a project. For example, the successful exchange of risk information between a program management entity and a system construction entity (usually a contractor) involves interoperability considerations.

Our focus on achieving interoperability in the acquisition process is between projects rather than within various organizations that participate in a single project. We do not anticipate new issues in applying the principles here to the context of a particular project. The characteristics

14. So must we consider interoperability between different upper management organizations?

of data and operations performed on that data to achieve interoperability are, we believe, equally applicable in an environment centered on a single project.

Is the basic tenet of the approach described here too limiting? In particular, the approach is based on developing *one* ontology from which *one* acquisition framework is derived, from which multiple acquisition models may be developed, consistent with that framework. This is a very top-down view. Is it not possible to have multiple ontologies and then integrate them? Or to have multiple acquisition frameworks and then integrate them? These two cases represent a bottom-up view. Our response is that having one ontology, from which one framework is derived, represents an optimal case. We recognize that there may be other general considerations (which got us to where we are in the first place), but we would like to solve a relevant subset of the general problem before introducing additional complications. Note, however, some of this wider problem is already present when we permit a user to specify some process (like requirements management) while the same process may be defined in an acquisition library. So the problem may already be here!

At what point in the approach should the decision about data elements be made? We have talked about ontologies and frameworks as a means for the expression of *concepts*. However, there is also a need to discuss the data attributes of those concepts. This could be presented in terms of the ontology or the framework; is one better than the other? Our preference has been to defer the discussion of data attributes to the domain ontology aspect of the framework. We say this for several reasons. First, there may be an ontology that could be reused that does not discuss details of data attributes. Second, it is not clear during ontology selection or development what the specification and description of the attributes should be. However, it is also possible that the ontology could include a specification of data and certain of its high-level concepts. For example, the DOLCE representation of quality could be used as a link to other data attributes.

3.2 Ontologies

What is the appropriate choice for the upper ontology for acquisition? There are many upper ontologies that can serve the purpose as the starting point for an ontology for acquisition. One of these, DOLCE, was previously mentioned. However, prior to selecting (or developing) an ontology we must be careful in the expectations we have regarding our choice. As we've indicated, there are many aspects to an ontology and they must be examined with some care so that the selected ontology will meet the goals of the work. We feel it would be relevant to identify a set of requirements for the upper ontology, no doubt in conjunction with other related ontology issues.

What is the appropriate choice of knowledge representation in an ontology? We earlier suggested that a frame-based approach may prove valuable as a means to capture knowledge.

However, this raises the question of the details that must be addressed. For example, there can be several types of frames. One type of frame may be used for description of knowledge content, while another may be used for assertions about that content. Still another type of frame may be used for configuration management of the data in a frame. The appropriate types of frames is a matter of special relevance, as it will play large in the amount and type of information that is available to assess interoperability in the acquisition process.

How can we handle the temporal aspect of the ontology specification? It is necessary that the ontology address the inherent temporal nature of the problem. There are two reasons for this. First, the elements of the ontology itself may change over time. For example, there may be a need for a new element to be added to the ontology at some point. The second, and more important reason, relates to the fact that knowledge encapsulated in the ontology, independent of structural considerations, is expected to evolve. For example, if the status of a project is an element of an ontology, it may be necessary to include a new value of the project status. The ontology is thus time dependent. A consequence that must also be addressed is that an associated formal reasoning system must also be capable of dealing with temporal considerations. The problems are more difficult, and more interesting!

How should an ontology deal with experiential knowledge? Ontologies are frequently regarded as presenting a single, current view of the knowledge they represent. Yet, as we have discussed, there would be value to achieving interoperability in the acquisition process if it were possible to account for experiential knowledge. Thus, we are faced with the problem of reconciling these different perspectives. One seemingly natural approach would be to embed experiential knowledge in a frame for some concept. However, it remains to be seen if this approach is viable. Note also the possible relation with a temporal view of the ontology, discussed above.

3.3 Acquisition Framework

What should be the scope of the acquisition framework? This question is a direct consequence of the first issue raised in Section 3.1, regarding the scope of interoperability (and inclusion of upper management). Expanding the degree to which the framework may be applied to a notion of a generic project (which could be applied to upper management function, for example) seems a viable notion, but must be examined.

What is the appropriate specification of an acquisition framework? Fundamental to this entire approach is the role of the acquisition framework. We believe it is through the framework that we can hope to achieve interoperability in the acquisition process. Thus, the proper specification of the framework remains one of the most important issues throughout the discussion. If that is the case, what exactly should be included in the specification of the framework? A start on a specification has been provided—through the use of activities, internal and

external events, system instances, entrance and exit criteria and so on—although additional material may be relevant to this topic. For example, should the concept of a policy be included the framework? How about *tasks* which are used to build activities?

What is the appropriate depth of information that should be encapsulated in the framework? It is one thing to say that a framework includes activities and so on. But it is another thing to state what the attributes of those activities are. We must balance the desire for versatility with the liabilities of overspecification when we consider the amount of detail necessary. This question is closely related to the structural approach for representation of knowledge. We touched on this point in our discussion of frames; see Figure 4 on page 8 for an example related to activities.

What are the applicable rules concerning the specification of the acquisition framework? We have earlier stated that the starting time of an activity must occur prior to the stopping time of an activity. This seems obvious, but there are other questions that are more difficult. For example, if an activity has an entrance criteria, should that same entrance criteria be an exit criteria (conservation of entrance criteria)? At first, this may be expected, but there are cases where this may not be the desired result. This simply illustrates that a specification of the rules applicable to an acquisition framework can be problematic.

What is the approach to management of change of an acquisition framework? We would expect very little change in the (upper) ontology, and perhaps some change in the domain ontology. The point of concern is with the framework, which specifies the different specific activities, events, artifacts, and so on. There is also the potential for change of the attributes associated with the concepts. For example, a new activity may be deemed necessary or some new attribute may be desired for some concept. Such potential changes can introduce volatility in the specification of the framework. Note that this problem is also related to the temporal view of the framework. In particular if one seeks to reason about previous elements of the framework, there may be serious difficulties.

3.4 Acquisition Models

How can the development of an acquisition model permit user tailoring? There is always the expectation that users will want to tailor their acquisition models for some project-specific reason. This can include modification of the basic acquisition elements (or their attributes), or additions to the acquisition elements. No matter how unappealing, the question of user tailoring must be faced.

It appears that tailoring of an acquisition model may be permitted *as long as* such tailoring does not affect interoperability. For example, suppose the specification of an activity includes an activity state. If the state of the activity (possible values might be *in_progress* or *complete*)

is not used in interoperability considerations, then adding a new value to this set should not introduce any problems. However, if the status of an activity is relevant to interoperability considerations, then we cannot just add (or change) values at will. Does this imply a partitioning of data with regard to how it relates to interoperability?

What does it mean for a specification of an acquisition model to be consistent? We have previously alluded to this. For example, the starting time of an activity must precede the stopping time of the activity. Other, similar statements can be made. But are we now heading toward a specification of a generic acquisition model? Notice that the acquisition framework does not specify the concept of an acquisition model; it simply provides items that can be combined to form a model. In the end, the model is a composition of elements from the framework, but there needs to be a way to state how that composition is developed. The rules under which that composition is valid must also be stated. Hence, should the acquisition framework include the concept of a generic acquisition model? And what would that be?

How can interoperability of acquisition models be addressed? In general, interoperability can be viewed in terms of the acquisition framework. Presumably then, the interoperability of acquisition models is implied as a consequence of the framework. If so, it places constraints on what information must be in an acquisition model in order to assess interoperability. It would be nice if the problem could be solved at the framework level, but it is not clear that is possible.

3.5 Acquisition Library

How can best practices be incorporated into a library of acquisition practices? There certainly is a goal to identify best practices so that others can apply those practices in their acquisition. There are a number of sources for best practices. These include community-accepted practices (e.g., CMMI) or industry standards such as those developed by the IEEE (e.g., for risk management) [Chrissis 03, IEEE 01].

Risk management is accepted as a standard practice for the management of a system. We have performed a brief assessment of risk management as discussed in the CMMI and the IEEE standard for risk management; see [Chrissis 03, IEEE 01].

Our brief results indicate that simply conforming to these practices does not assure interoperability.[15] That is, two projects can conform to these approaches yet not interoperate.[16] In our

15. Whether these indicated best practices were intended to deal with interoperability as part of their scope is somewhat out of the question. The fact is that they do represent community consensus and will therefore be looked upon favorably and are likely candidates for use in the community to achieve interoperability, despite their possible shortcomings.

view, the practice of risk management, as presented by the IEEE standard, is not sufficiently specified to achieve interoperability for multi-project risk management [IEEE 01].

Given the preceding assessment what should be done? Should these best practices be further extended so that they address interoperability? This is a lot of work!

We also know, from experience in the standards community, that standards are developed with one of the underlying principles being resolution of tension among those who develop a standard. On the one hand, there are those would like a general specification of a standard so that they can claim conformance to that standard (and extend the standard to give them product-unique features). In contrast, there are those who would prefer a standard that included more detail, where certainly one of the goals may be a tighter form of conformance—or even interoperability!

Thus, despite the prevalence of best practices, we wonder if sufficient information is specified to assure interoperability. In some sense, existing practices can be viewed as one-dimensional in that they take the perspective of an acquisition project. We believe we need to deal with a second dimension, that of multi-project acquisition, in order to achieve interoperability.

Who owns the specification of a library of best practices in terms of the acquisition framework? If a best practice is not sufficiently specified in order to achieve interoperability what must be done? Clearly, some modification to the best practice must be put into place. But who will perform this task? One might argue that this is a community-based responsibility, and that standards for interoperability among various practices must be created. Of course sooner or later we may run into standards-development organization conflicts. This particular question is one of further development and transition of the work specified here and should be examined in such a light. Perhaps profiles for interoperability might help.

3.6 Integration Process

Is it a viable assertion that the integration of multiple projects may itself be represented as a project? We made this assertion in Section 2.6. If true, then it is possible to have the same expressive structure (i.e., framework and model) for the specification of a project *or* for a "project" whose purpose is the integration of multiple projects. The possible gain in reuse is significant. Further, it means we can apply this notion recursively to arbitrary levels of projects. This would be another significant benefit, provided through the reuse of structures and operations on those structures. It should solve, in essence, the question of scope of organi-

16. Here is an obvious example. Two projects have the notion of a risk status (never mind for the moment what this might be). But one project uses a color scheme of red, yellow, and green for the values of a risk status. Another project uses values one through five for risk status? How can these projects interoperate?

zations that participate in different facets of acquisition (recall, for example, our earlier issue about the role of upper management). Hence, how generally can a project be specified such that it can serve the different roles necessary to address acquisition?

What is the relation between integration of acquisition models and the corresponding integration of (a domain) ontology? Simply stated, we view an ontology and a model as very similar. In other words, while they may have different structural representations, they have (largely) identical semantics. We view this in a positive light, as it allows us to view the same thing in different ways.

What is the role, if any, for a language in discussing integration and interoperability of projects? Here lies the crux of one of the important problems. Underlying the question of a language for discussing integration is a deeper examination of the term. For example, what does it really mean to integrate two acquisition projects—regardless of what they may happen to be—in order to achieve interoperability? We propose that this question be addressed in the context of a formal approach; see further Section 3.8.

3.7 Experiential Knowledge

Is it a viable premise that experiential knowledge can be expressed in the context defined by the concepts of an acquisition framework? We accept the idea of an experience factory as being potentially valuable and believe it has a place in this work. But we are greatly concerned that the experience gained must have some associated context. We have too many times used the analogy of just "throwing darts at a wall" to convey the implication of a lack of context for experiential data. We need to examine in more detail this question regarding the relation between the structure of an acquisition framework and that of a knowledge organization as presented in an experience factory.

What is the appropriate representation of experiential knowledge? The assertion that experiential knowledge may be described in terms of the concepts of the acquisition framework is a structural view. There remains the manner in which experiential information may be represented. We are close to the analogous question of relating experiential information in an ontology. For example, can experiential knowledge simply be represented as a slot in some frame used to describe some concept (see Section 2.1) or is it otherwise special, and if so, how? The very nature of experiential knowledge forces us to take a temporal view of a framework or an ontology.

What is the relation between experiential knowledge and an acquisition library? It is, perhaps, natural to assume that such a relation exists. In particular, one might expect that experiential knowledge about some component of an acquisition practice would be a part of that practice. If there is a practice related to risk management, one might expect that experiential

knowledge about that particular practice would also be part of the acquisition library. This point seems to be the proper perspective, but requires further assessment to understand the full implications.

How can experiential knowledge provide interoperability benefit to a participant in an acquisition context? There are questions about structure and semantics of experiential knowledge. Here we are concerned with this question: If you had an experience factory how would you use it? Many types of analyses are possible, such as linguistic analysis to develop a greater understanding of the experiential knowledge. A key, from our perspective, is that an experience repository should not simply be passive, but capable of interacting with a user, possibly autonomously. Much work lies ahead in this area, but first it is better to understand the problem!

How can the management of experiential knowledge deal with possible contradictions? If there is experiential knowledge that "Product X is great." there may also be experiential knowledge that "Product X is terrible." A clear inconsistency is present. Naturally, one approach is to present all experiential knowledge to a user. Another is to seek to understand, on a deeper level, the nature of possible inconsistencies. We view such a task as quite challenging, and possible inconsistencies may be admissible, independent of any attempt to resolve them.

How can experiential knowledge be used by other projects in a predictive manner? This is the other side of the coin. While experiential knowledge has value to a project, it would have greater value if it could be productively used by other projects. We would like to be able to develop a formal approach that supports reuse of experiential knowledge in a predictive manner, along the lines of reasoning by analogy.

Some work has been done in the subject of predictive risk management,[17] exploring the question "Given a set of risks to one project, how can others use this information to identify risks for their projects?" There are many issues here, some of which have been identified above, such as representation of risk data and the ability to share models for risk management. In a sense, we have returned to the question of integration of ontologies and models.

3.8 Role of Formalism

What is the appropriate formal specification language for use in this work? There are many formal languages available, typically a variant of predicate logic. However, recognizing the need to account for the explicit temporal dependence, as well as reasoning over a specification, we are beyond a simple predicate calculus. We are considering a formalism that is

17. B. Craig Meyers and Ira A. Monarch, unpublished report.

intended to account for the dynamic nature of a system and that may have application to this work.

What is the specification of acquisition such that interoperability may be achieved? While we have initiated a formal specification of an acquisition *framework* (and hence models derived therefrom), the problem remains of how interoperability in the acquisition process should be specified. Some general approaches to interoperability have been addressed. We have not, however, specifically examined interoperability in an acquisition context.

Let us speculate a moment; we would like to be able to develop a formulation that contains statements such as

> Let P denote a set of projects that are producing systems that are expected to interoperate. The set of projects can interoperate with respect to some external event e iff all projects are capable of processing the event e in such a way that the schedule of P, denoted S(P), is satisfied.

The above is an example of what we believe a formal specification of the acquisition process should contain.[18] The fact that it is mathematical does not bother us. It is from the mathematical perspective that we can deduce practical statements. In the above example, we might desire, for instance, that no external event creates a lateness condition to the overall schedule.

What would a formal specification of an experience factory contain? This question has merit in its own right, but is especially relevant in its relation to an acquisition framework. We would also like to be able to state the conditions under which an experience factory is *well formed.* Such information can be used to assess the work of a particular project.

3.9 Pragmatic Concerns

Can the philosophy of interoperability in the acquisition process sufficiently demonstrate value to the acquisition community? The acquisition community is always overburdened and busy responding to issues of the moment. Further, as is well known, much of that community has a project-centric focus, as opposed to a larger, system-of-systems focus where interoperability is paramount. The injection of technologies in this domain faces an uphill battle. Key to its success will be the ability to demonstrate value to the end user.

How much can automated support for interoperability in the acquisition process be provided and used? Certainly the role of tools will be important to an organization that is engaged

18. More precisely, the example cited could be described as satisfiability of a schedule for multiple projects with respect to an external event. No doubt other statements could, and should, be developed. *We believe that development of such statements is fundamental.*

in the day-to-day process of acquisition. But exactly what are the desired tools? There are many tools that are available that can be used in this effort.This variety of available tools eventually brings on the issue of achieving tool integration. Some work has been initiated on a tool to support interoperability in the acquisition process along the lines discussed in this report.[19]

More important is the possibility of inter-tool interaction so that some aspects of an acquisition can be performed without human intervention. We envision a tool that can initiate interaction with a user to help in achieving interoperability. We are all well aware of the social issues that limit multi-organization, multi-domain interoperability. Luckily, computers do not have such restrictions.

What is the appropriate transition path for the future of this work? The development and application of any new technology approach requires planning, and implementation, with care. If the ideas expressed in this report are deemed viable, then the transition for this work should be addressed sooner, rather than later.

3.10 A Matter of Philosophy

There are different ways of looking at the material we have presented. The philosophy of how one views obtaining interoperability in the acquisition context is important.

On the one hand, it is possible to take the perspective that "all programs should do acquisition the same way" and that what we are really after here is a standardized, rote way of doing acquisition. It could, therefore, be viewed with disdain as yet another regulation which already overburdened acquisition projects must deal with.

The other hand counters the above argument in that we are trying to provide a means for an acquisition project to succeed in an interoperable environment. We believe that success is based on the use of the acquisition frameworks, models, ontologies, experience of others, and yes, even mathematics. There probably is the need for standardization, but it may not be a need for standardization of *everything*.

Where lies the answer? No doubt somewhere between the above extremes. But interoperability is a horse of a different color. Successfully achieving interoperability is more than simply the success of a single project; it is the collective success that we are all after.

19. This work was performed at Carnegie Mellon University by Nate Watterson.

4 Usage Scenarios

In this section we present two brief scenarios to demonstrate the use of the proposed approach to achieve interoperability in the acquisition process. An additional usage scenario is included in the appendix.

Our intent here is to focus on conceptual aspects of the problem. Of special importance is the identification of issues relevant to achieving interoperability in the acquisition process. Resolution of the issues is required in order to understand the expected behavior of the acquisition system. Although we believe that automated tools can significantly aid achieving interoperability in the acquisition process, first we must understand the problem and its solution.

4.1 Approach

Each usage scenario is organized along similar lines and includes

- background: A brief statement of the scenario will be presented. The scenarios are based on our experience in dealing with acquisition programs; in at least one case, however, the scenario will be based on a *Gedanken* experiment, in which a scenario is generated as a result of a thought experiment.
- acquisition model: An approach to the usage scenario will be presented in terms of an acquisition model; see Section 2.3. For brevity, we omit the steps leading to the development of an acquisition model (ontology and framework) to focus on a *particular* model.
- discussion: We will include a discussion of the acquisition model approach to the scenario. It will also include consideration of the way the scenario is treated in typical acquisitions.

We will also illustrate the role that an acquisition library (see Section 2.4) and experiential knowledge (see Section 2.6) may play in the approach to the scenario development. Finally, we will also mention cases where formalism can come into play.

In the scenarios to follow, we will assume that the task of developing the ontology and framework has been completed. We do this to present the reader with examples of the application of the concepts regarding interoperability in the acquisition process. However, we warn the reader not to be lulled into thinking that the development of the (domain) ontology and framework are simple.

We emphasize that the material to follow, especially discussion of the acquisition model for each scenario, is but a mere outline of the solution approach. There is both breadth and depth to the solution that would have to be developed for a full treatment of the problem. The examples presented here will be based on only two acquisition projects, a choice made only for simplicity in presentation. Our intent is to give the reader a sense for how the concepts of interoperability in the acquisition process can be used, rather than illustrate their detailed application (including the role of formalism).

4.2 Requirements Management

Background: For this case, we assume there is a project that is responsible for the development of requirements which are then levied on systems being developed by other projects. This is characteristic of a top-down approach to a system-of-systems (SoS) acquisition. Because of the SoS perspective we will illustrate the possible conflicts in requirements and how they might be managed across multiple systems. All SoS acquisitions must be able to manage possible requirements conflicts. We believe that managing the requirements in a top-down strategy will be effective in managing such conflicts.

Acquisition Model: There are many requirements management processes, such as one specified by the IEEE [IEEE 98a] and one in the CMMI literature [Chrissis 03]. Requirements management is also treated in another IEEE Standard [IEEE 98b], in the context of software acquisition. For this example, we assume a process that only includes activities for

- requirements elicitation
- requirements analysis
- requirements validation

The assumed acquisition model is shown in Figure 13. We show two projects, A and B, that are developing systems. One of these projects is performing its acquisition in a waterfall manner, but the other is being performed in a spiral manner. In the center of the figure we show an SoS project that is responsible for integration of the efforts of the two other projects, such that interoperability is obtained. Naturally an SoS project must perform a myriad of other activities, but here we focus only on that aspect related to requirements management. Notice that the SoS project has started after one of the other projects, but before another. Notice the interaction between the SoS project and the two local projects in terms of how validated requirements are provided to the local systems.

It is natural to partition the requirements into two categories for requirements that are applicable to the following:

- the system of systems; presumably, these requirements also account for interoperability among systems.

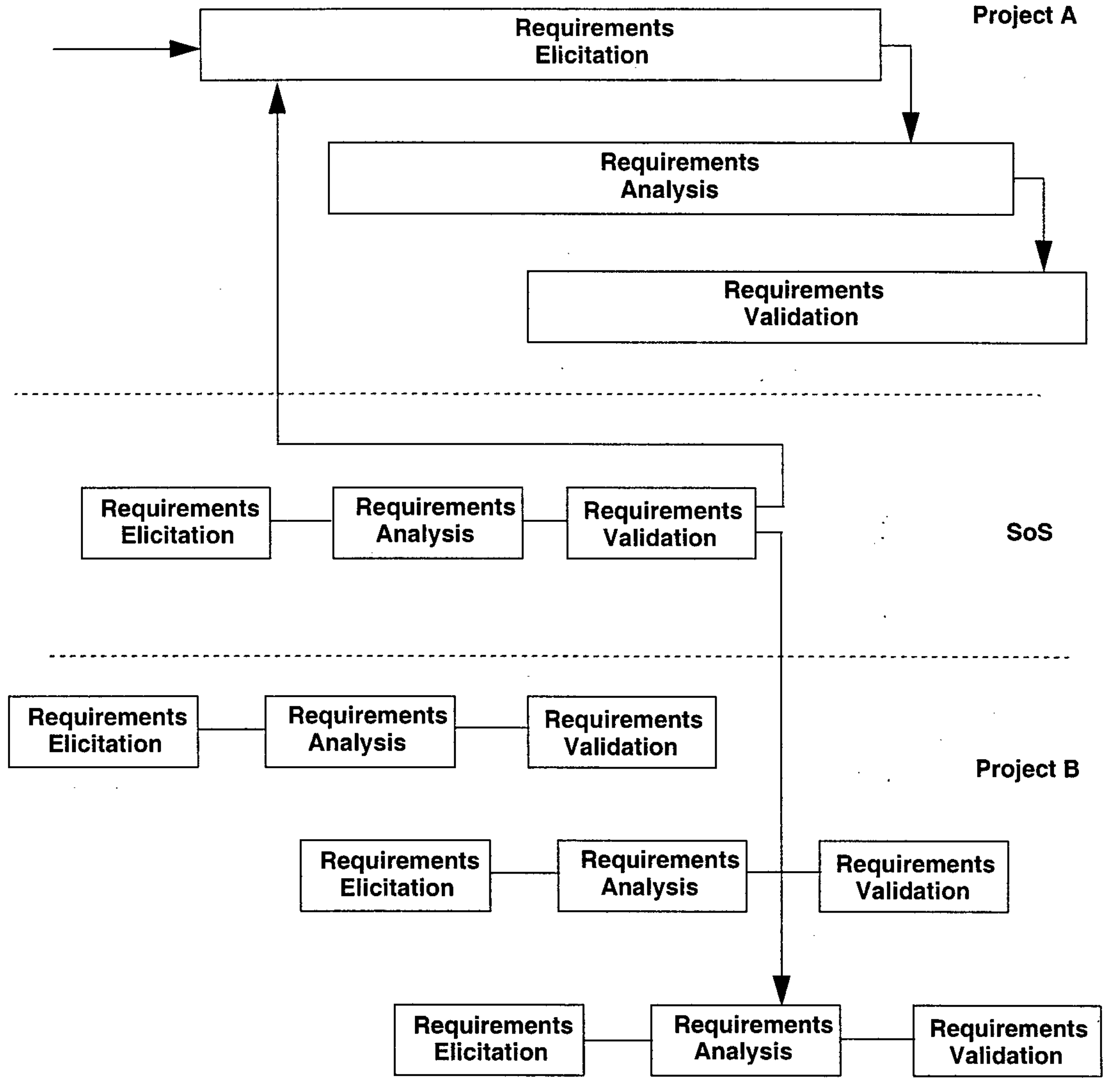


Figure 13: Acquisition Model for Requirements Management Scenario

- an individual system; presumably the requirements are consistent within a given system.

This approach to partitioning requirements between an SoS view and a local view is important, and we show a relation among the requirements in Figure 14. While the process of partitioning the requirements is not explicitly accounted for, it may be performed as part of the management of requirements between the SOS and the local projects. Some relevant remarks for this figure include:

- All of the SoS requirements are implemented by at least one individual system. It may also be possible for an SoS requirement to be implemented in multiple systems. This may arise

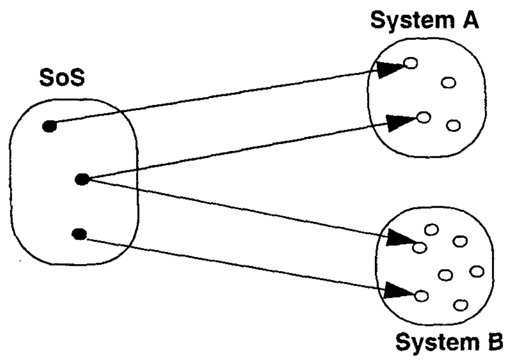


Figure 14: Partioning SoS Requirements to Local Systems

from the fact that an SoS requirement is expressed as a composition of system requirements. Or, it may be the case that, for fault-tolerance reasons, the same SoS requirement is expected to be implemented in multiple systems.

- Each individual system has system-specific requirements to meet (presumably independent of any other system).

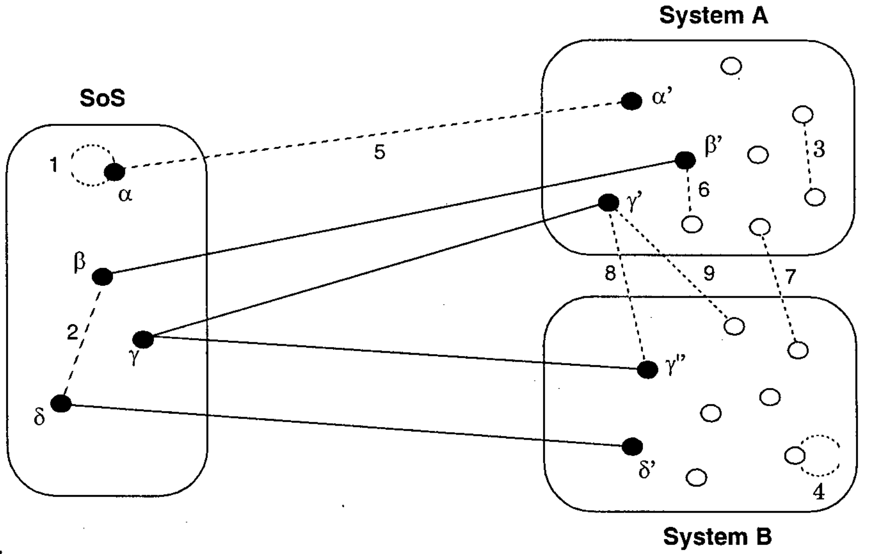
The mechanism by which requirements are partitioned between SoS and system-specific projects is fundamental. It will bear on how the overall requirements are managed.

Discussion: Even a simple approach as indicated for requirements management has some interesting issues. Regarding the requirements identified as applicable to the SoS, we raise the following questions:

- Are the activities of *Requirements Analysis* and *Requirements Validation* the sole responsibility of the SoS project? Or, do the other projects participate?
- How are the requirements that are applicable to the SoS provided to the other projects? In Figure 13 we have assumed that a requirement must be validated by the SoS project before it is provided to the other projects. There is a need to make sure that an SoS-generated requirement does not conflict with a system-specific project. To determine this assessment, some form of requirements analysis must be done by the individual projects which consume requirements from the SoS project.
- Should the SoS project have a role in the requirements validation activity performed by other projects? Presumably, the other projects must implement the SoS requirements. Is knowledge of the validation process deemed within scope of the SoS project? That is, if the individual projects must validate SoS-generated requirements, then the SoS project may want to know details of this effort, such as exit criteria for the validation activity and so on. Also, how are the exit criteria for validation determined, and, in particular, by which project?

- How are possible conflicts identified, validated, and adjudicated between SoS requirements and project-specific requirements?
- How are possible conflicts identified, validated, and adjudicated between individual project-specific requirements?

A view of the possible conflicts among requirements is shown in Figure 15. We have assumed a project responsible for the SoS requirements, which are then implemented by two other systems.



Notes:

- SoS requirements are denoted by Greek letters. Particular system implementations of those requirements are denoted by primed Greek letters.
- Cases of possible requirements conflicts are indicated by numbered dashed lines. The various cases are discussed in Table 3.

Figure 15: Sources of Requirements Conflict

Further discussion of the possible conflicts in requirements is shown in Table 3. The discussion there is tied to the cases identified in Figure 15.[20]

Notice that the choice to distribute requirements among various systems and to distinguish types of requirements (i.e., SoS requirements and system-specific requirements) is accompanied by the need to distribute and manage potential conflicts among both requirements and the systems responsible for their implementation. In fact, when we consider the example of possi-

20. An interesting question is how the problem of requirements management across multiple projects is dealt with. This question also involves the approach one takes to the ontology!

Table 3: *Discussion of Possible Requirements Conflicts*

Case	Description
1	An SoS requirement (α) may be inconsistent. This is a form of self-conflicting requirement.
2	There may be a conflict between two SoS requirements (β and δ)
3	A local system may have a conflict among local, system-specific requirements. This does not involve a conflict involving an SoS requirement.
4	A local system may have an inconsistency in a local requirement. This does not involve a conflict involving an SoS requirement.
5	An SoS requirement (α) may conflict with an intended implementation of that requirement (α') by some local system.
6	A conflict may exist between an SoS requirement allocated to a local system (β'), and a local requirement.
7	A conflict may exist between two local, system-specific requirements, implemented by different local systems. This does not involve a conflict involving an SoS requirement, but could affect interoperability.
8	An SoS requirement (γ) has been decomposed into two requirements, each of which is implemented in different local systems (γ' and γ"). The system-specific implementations of those requirements may be in conflict.
9	A conflict may exist between an SoS requirement implemented in one system (γ'), and a local requirement implemented in another system

ble requirements conflicts, it may warrant a reconsideration of the basic elements of the acquisition model.

Consider Figure 16 which shows an SoS project and only one local project. We have shown a new activity of *Requirements Conflict Management* for both the SoS project and the local project. Some of the possible conflicts identified may be resolved by the SoS project (cases 1 and 2) or by the local project (cases 3 and 4). However, there are possible conflicts, indicated in the center of Figure 16 that will likely require joint effort for their resolution. In particular, we are referring to cases 5-9 as discussed in Table 3.

We have shown various cases of requirement conflict in this example. More importantly, we suggest a need to define an activity for *Requirements Conflict Management* that is responsible for the identification and adjudication of possible conflicts in requirements. It is important to note that such an activity is not highlighted in a system-specific view of a project (where requirements conflicts may be addressed as part of the activity for requirements analysis). Here, a project-centric view will not suffice.

Again, we see the value of developing a specification of the problem, in terms of an acquisition model. The development of the ontology issues and acquisition framework have not been shown. Some aspects for consideration might include

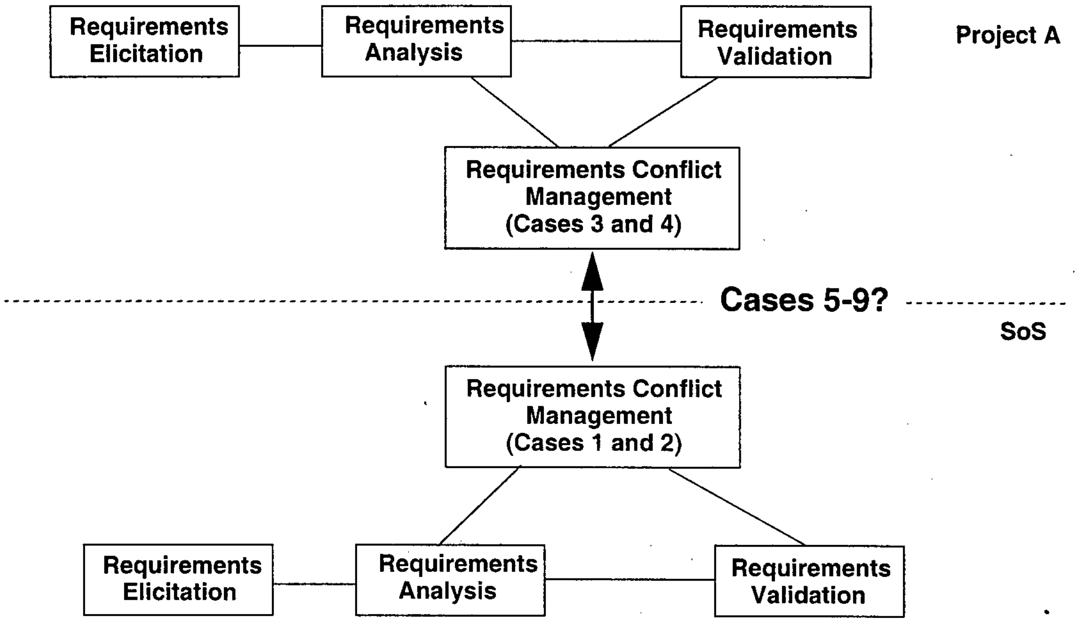


Figure 16: Revised Acquisition Model to Account for Requirements Conflict Management

- How are requirements handled in the ontology? In some sense, a requirement is a description of a property of a system element or the system itself. Recall, from our earlier discussion (see in particular footnote 8 on 11), that we view requirements as fundamental to the discussion of interoperability in the acquisition process. But look at Figure 2 on page 7 where we illustrated the *DOLCE* ontology; where do requirements fit in that scheme? Is a requirement a quality? Or is it an assertion about a concept at a particular time? We view the resolution of questions such as these as fundamental, as it has implications for the overall knowledge structure relevant to acquisition.
- Given that the concept of a requirement is present in the ontology, the notion of a requirement will have to be in the framework. The acquisition framework may have to provide an operation that indicates if two (or possibly more) requirements are in conflict.

Finally, there is a very important point that applies to this scenario, as well as all others to follow. We must be careful to distinguish *what* is done (such as an activity) as opposed to *when* it is done. These aspects are accounted for in the framework. While elements of the framework may have temporal characteristics (such as a start and stop time), the framework itself is silent with respect to *how* the elements are composed. Such temporal composition of elements of the acquisition framework is a life-cycle matter. In this example, one project assumed a waterfall approach while the other performed a spiral approach. Both of these approaches can be specified using the acquisition framework.

4.3 COTS Product Upgrade

Background: Assume there are two projects producing systems for their individual acquisitions and that these two projects use COTS products. Also assume there is a third project which is responsible for the acquisition of COTS products for *both* projects. What happens when there is an upgrade to a COTS product?

Acquisition Model: From the perspective of an acquisition model it is natural to represent the upgrade of a COTS product as an external event. Various attributes of an external event can be identified; an example was shown in Figure 2 on page 7.

Many possible activities related to the use of COTS products may be specified; see Meyers for some general discussion and Albert for a more detailed example [Meyers 01b, Albert 02]. For the project responsible for the acquisition of COTS products we will limit the discussion to activities related to COTS product evaluation, procurement, and configuration management. For the projects that are using the COTS products, we will only consider the activity of system integration.

The resulting partial acquisition model is shown in Figure 17.

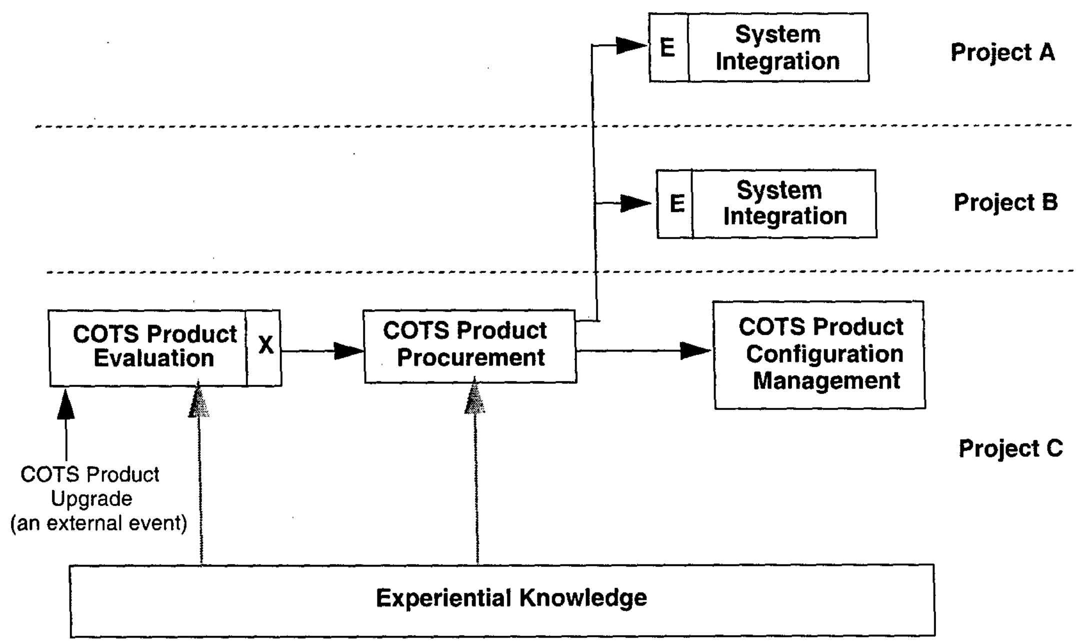


Figure 17: COTS Product Upgrade

Discussion: The main thread in Figure 17 is that in which an upgrade to a COTS product becomes available (related to an external event from the perspective of an acquisition model),

is evaluated, and then procured. The selected COTS product is then provided to both projects for incorporation in their system integration efforts.

Several issues are relevant here, including

- What is the role of the consumer projects with regard to the development of the *COTS Product Evaluation* activity? Should this activity be jointly performed by both the project acquiring COTS products, as well as the projects consuming COTS products?
- What is the role of the consumer projects with regard to the development of the exit criteria for the *COTS Product Evaluation* activity? Should these be jointly developed? There may be those who would like the criteria to be specific to *their* project.
- What is the relation between the exit criteria for selection of a COTS product and the entrance criteria for using that product? Must the criteria remain unchanged (conserved) or can a consuming project change them?
- What happens with regard to the timing of a COTS product upgrade becoming available versus the system integration activities performed by the various projects that may use that product? How can the consuming projects adjust their schedules to accommodate perturbations caused by new COTS products?

We have also shown experiential knowledge in Figure 17, discussed earlier in Section 2.6. Recall that one of the aspects of the assertion regarding use of experiential knowledge was that it be based on the elements of the acquisition framework. One of those elements is an *artifact,* and a COTS product is a particular artifact. There may be experiential information related to knowledge of

- the particular COTS product that has been upgraded
- evaluation activity for COTS products
- procurement of the COTS product: This might include information about the vendor (a form of organization), and there may be information about the license policy for this product.

This small scenario where one project performs activities on behalf of another is an example of what we have termed in past work as part of a *common acquisition infrastructure* [Meyers 01b]. In other words, there may be certain elements of acquisition—such as COTS product evaluation—that are performed by one project, on behalf of other projects. We are unaware of current acquisitions being structured such that they can use the concept of an acquisition infrastructure. For example, the evaluation of a COTS product could be done by one project on behalf of other projects. We believe it is worthwhile to assess the use of this concept as a means to achieve interoperability in the acquisition process.[21]

4.4 Summary

We have provided two scenarios related to interoperability in the acquisition process. Our purpose was to illustrate the approach taken in this report, particularly through the use of an acquisition model. There is ample illustration in the scenarios for how interoperability in the acquisition process might apply. But there is also ample illustration of just how difficult it may be to obtain, thus demonstrating the problem of achieving interoperability in the acquisition process.

One additional benefit is worth noting: *The process of specifying the intended acquisition is of considerable merit in its own right.* This represents the first step to understanding the behavioral aspects of the overall acquisition system for which we strive. Shouldn't acquisition be treated on such a basis?[22]

21. We will add one point. One of the points of departure for developing an architecture for an operational system is to split it in two parts: *Applications* which run on top of an *infrastructure*. Could that approach apply to an acquisition system? Are there political and funding impediments that would limit the ability to apply this concept? How would programs react to this concept, given that it would represent some loss of control? There is more here than meets the eye!

22. We ask the reader to ponder about developing a rigorous *requirements specification* for an acquisition project that would account for all the things necessary to achieve an acquisition, such as activities, events, and so on. Then, having solved this problem, revisit the task for an acquisition *system* and incorporate interaction with other projects. Hopefully, this challenge will resonate with the approach put forward here.

5 Summary

We wish to develop a means for achieving interoperability in the acquisition process. We propose that developing interoperability of program management and system construction will bring successful interoperability within closer reach. We are after a larger picture.

It is the use of a well-defined acquisition framework, and models derived from that framework, that affords us a mechanism to achieve interoperability. There is continuing tension among project teams in regard to interoperability. We witness a struggle between "I'll do it my way" and "You'll do it the same way." The approach we have described attempts to preserve the local view of a project, but only where appropriate. In the end, we want to promote a view that permits effective acquisition. In fact, a goal of this work would be realized through a more mature approach to acquisition.

We close on a philosophical note. We expend great intellectual effort on the specification, development, and operation of computer systems. That effort has led to great rewards in the systems that we have developed. We think the treatment of an acquisition *system* should employ the same skills and the same approaches that are associated with an operational system. It deserves no less.

Appendix Usage Scenario for Multi-Project Reuse

In this appendix we provide an additional usage scenario that deals with reuse across multiple projects. This example will go into a bit more depth than the examples of Section 4. The elements of the approach here are the same as those discussed in Section 4.

Background: For this scenario, we assume there are a number of projects developing products. We further assume there is a project which is responsible for the integration of the products (recall the discussion of Section 2.5), adding the necessary glue code, and then producing a system that is the composition of the work done by others.

Acquisition Model: For the simple case of two projects producing reusable products, the relevant subset of the acquisition model is shown in Figure 18. Here we show product development activities in two different projects, each of which has a specified set of exit criteria (X). At the center of the figure we show the integration project and the activity for product integration. Also shown are the entrance criteria (E) for the activity of product integration.

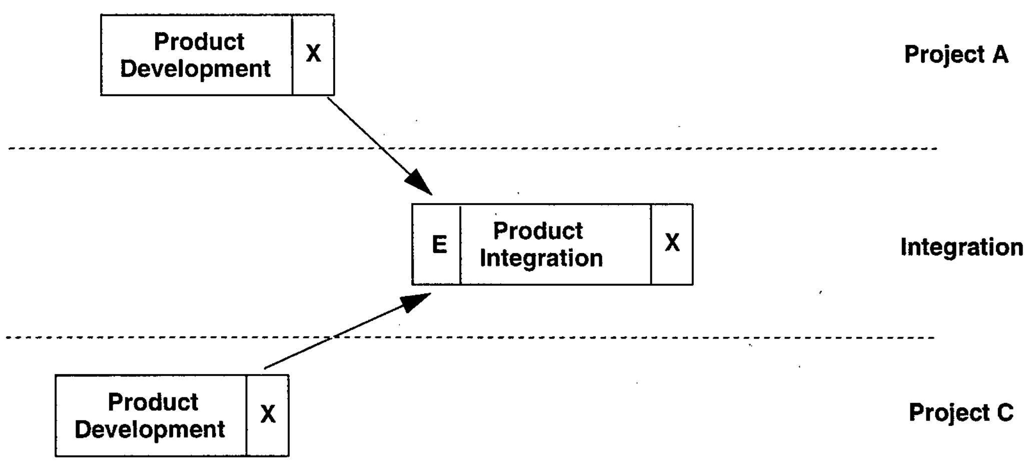


Figure 18: Basic Multi-Project Reuse

Discussion: Of particular relevance is the relation between the exit criteria for the projects developing products and the entrance criteria for the integration project. Some considerations include

- If the exit criteria for the product development activities are more stringent than the entrance criteria for the product integration, all appears well.
- In contrast, if the exit criteria for the product development activity are less stringent than the entrance criteria for the product integration, a host of problems may emerge.

How can the approach here help? First, one element of an acquisition model is an activity that also includes entrance and exit criteria. One of the requirements for an activity to be initiated is that the entrance criteria must be satisfied. Such an assertion would come from a formal specification of the interaction of the components of an acquisition model.

In several cases with which we are familiar there is a lack of communication between the integrator and the developers with respect to acceptability criteria. In particular, the approach to integration of products developed by others is largely *ad hoc*. Why? Simply because such communication involves crossing programmatic boundaries (or worse, the boundary that may include the system development).

In fact, we would go so far as to suggest an alternative to Figure 18 and instead, consider Figure 19. Some of the aspects of product integration shown in Figure 19 include the following:

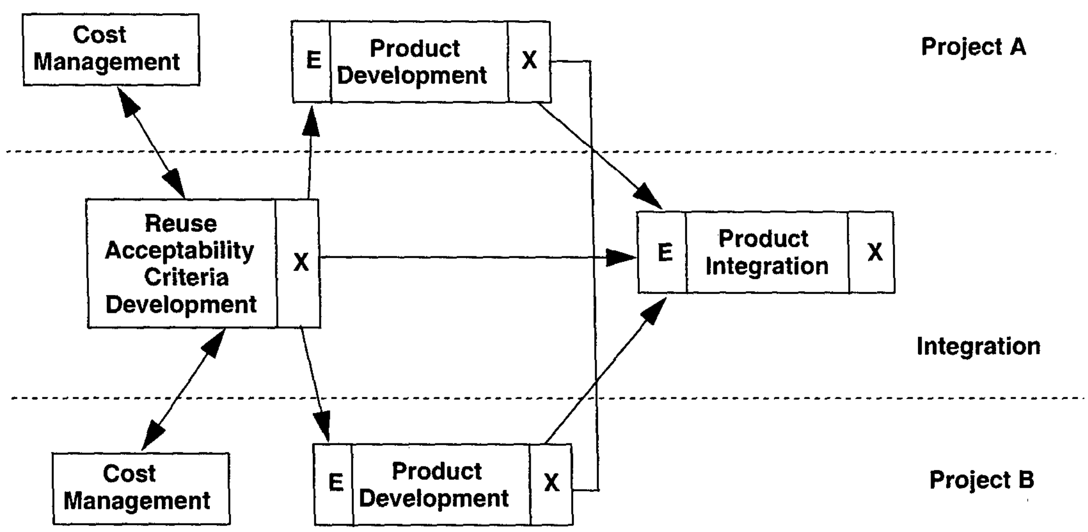


Figure 19: Enhanced Version of Multi-Project Reuse

- The integration project now includes an activity whose purpose is the development of acceptability criteria for reusable products. Deciding which organizations participate in

this activity is problematic. Ideally, one might expect that each organization that contributes reusable products should be a member.

- The reuse criteria now become the entrance criteria for the projects developing reusable assets. Presumably, these criteria would be conserved by the product developers so that they become exit criteria for the development of the reusable products.[23]
- The exit criteria for the product development now become the entrance criteria for the product integration. Upon meeting these criteria, the reusable products are then delivered to the product integration, as indicated in Figure 19.
- There is a connection between the activity of *Reuse Acceptability Criteria Development* and *Cost Management* for projects that will produce the reusable products. This simply reflects the fact that there may very well be a cost that must be borne by some project—which one, who pays, and who decides?—to provide products that conform to the required acceptability level of the project integrating those products.

Let us now add to this scenario consideration of schedule management. We will include such an activity for the two development projects as well as the integration project. The revised acquisition model is shown in Figure 20. Several items to note about the inclusion of an activity for schedule management include:

- Each development project now interacts with its own schedule management activity. Such an activity would be responsible for the schedule of the individual projects.
- The integration project also includes an activity for schedule management.
  - An entrance criterion for the schedule management for the integration project is shown. It is plausible to expect that schedule estimates provided by the development projects have some degree of fidelity.
  - An exit criterion for the schedule management is also shown. We might expect that such criteria specify the conditions that must be satisfied in order that an "official" schedule be produced.
- There are interactions between the development projects and the schedule management activity of the integration project. This is expected, and necessary, so that the integration project can maintain a relatively accurate schedule.
- There is an interaction, in the context of the integration project, between the activities of *Schedule Management* and *Product Integration*. This interaction is again necessary to maintain the schedule for the integration project.

23. Within the context of an acquisition activity, it may be possible to have exit criteria that are modified from the entrance criteria. The simple case of "conservation of entrance criteria" to produce exit criteria then becomes necessary.

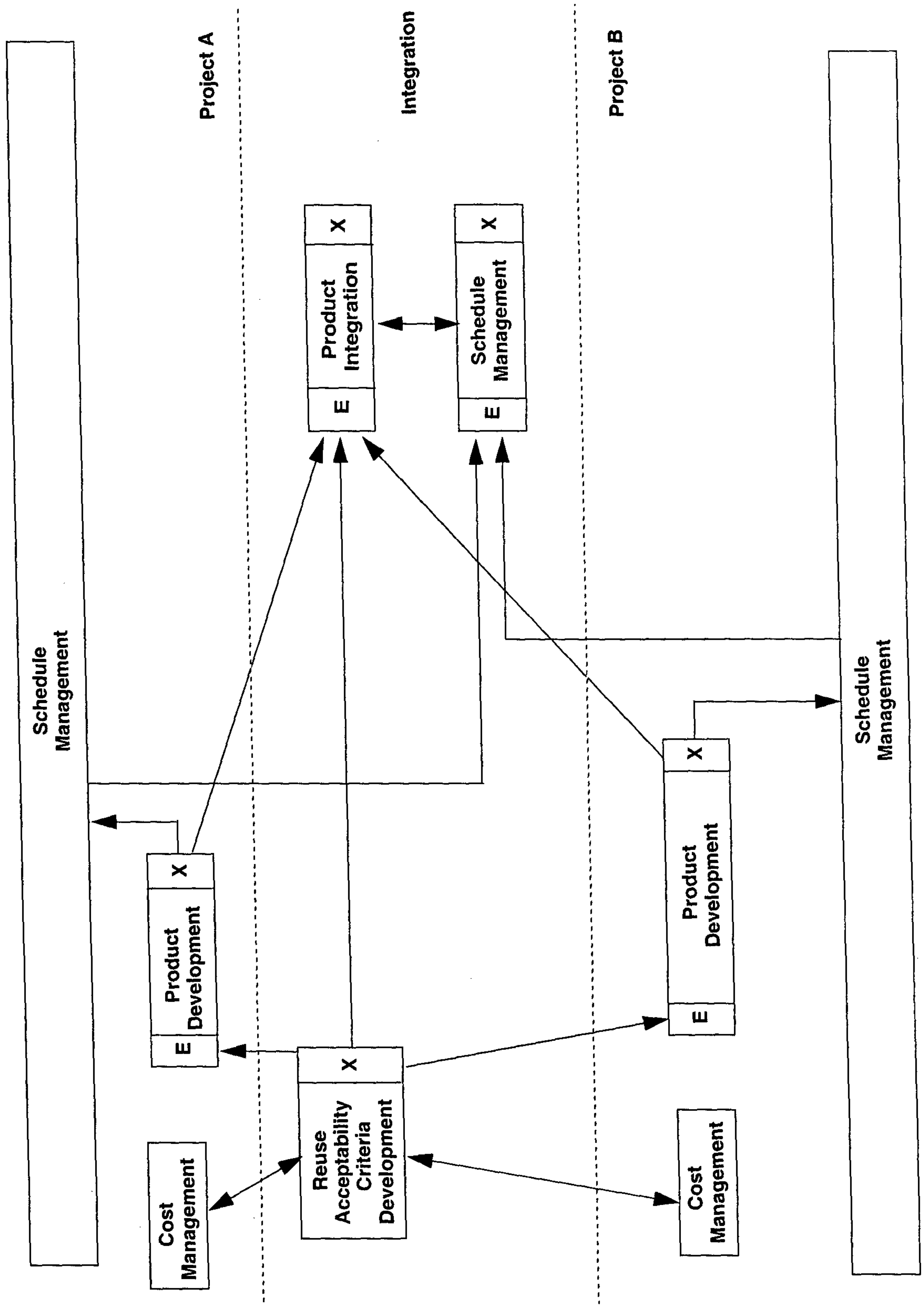


Figure 20: Inclusion of Schedule Management

Including activities for schedule management demonstrates the need for interaction between the projects involved. We have not shown an interaction between cost and schedule activities, although that would no doubt become necessary.

Also not shown in this example are any possible events that may occur. One might expect an (internal) event that would account for a variation in schedule by one of the projects producing products which are then integrated. The presence of this event can then be used to initiate any necessary response by the integration project.[24]

This scenario has illustrated some of the considerations in reuse of products across multiple projects. As implied from the foregoing discussion, we are faced with a difficult problem. The difficulty is manifest in the coupling between the system integration effort and the work of the product developers. This coupling can take many forms, hinging on criteria for acceptability of reusable products. Ultimately we come home to cost (and schedule) considerations, further complicating the problem.

24. It is interesting to speculate on the conditions that should cause an internal event to be initiated by one of the development projects with regard to schedule management. If a schedule variation of 5% is estimated should the event be raised? How about 20%? Some might respond that any variation should cause an event to be raised to the integration project. But there is the question of the fidelity of the schedule estimates that plays a role in this discussion. In the end, the estimated variation should be a configuration parameter to the raining of the event.

References

[Albert 02] Albert, C. & Brownsword, L. *Evolutionary Process for Integrating COTS-Based Systems (EPIC),* (CMU/SEI-2002-TR-005, ADA408653). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu /publications/documents/02.reports/02tr005.html.

[Basili 94] Basili, Victor R.; Caldiera, G.; & Rombach, H.D. "The Experience Factory,"469-476. J. J. Marcinik (ed.) *Encyclopedia of Software Engineering 1*, New York, NY: John Riley and Sons, 1994.

[Boehm 88] Boehm, Barry W. A Spiral Model of Software Development and Enhancement, *IEEE Computer 21*, 5 (May 1988): 61-72.

[Chrissis 03] Chrissis, Mary Beth; Konrad, Mike; & Shrum, Sandy. *CMMI: Guidelines for Process Integration and Product Improvement.* Boston, MA: Addison Wesley, 2003.

[IEEE 98a] Institute of Electrical and Electronics Engineers. "IEEE Recommended Practice for Software Requirements Specifications," *IEEE Standard 830.* Piscataway, NJ: IEEE Standards Association, 1998.

[IEEE 98b] Institute of Electrical and Electronics Engineers, "IEEE Recommended Practice for Software Acquisition", *IEEE Standard 1062.* Piscataway, NJ: IEEE Standards Association, 1998.

[IEEE 01] Institute of Electrical and Electronics Engineers, "IEEE Standard for Software Life Cycle Processes—Risk Management," *IEEE Standard 154.* Piscataway, NJ: IEEE Standards Association, 2001.

[Levine 03] Levine, Linda; Meyers, Craig B.; Morris, Ed; Place, Patrick R. H.; & Plakosh, Daniel. *Proceedings of the System of Systems Interoperability Workshop (February 2003),* (CMU/SEI-2003-TN-016 ADA416429). Pittsburgh, PA: Software Engineering

Institute, Carnegie Mellon University, 2003. http://www.sei.cmu.edu/publications/documents/03.reports/03tn016.html

[Masolo 03] Masolo, Claudio; Borgo, Stefano; Gangemi, Aldo; Guarino, Nicola; & Oltramari, Alessando. *Ontology Library*, Version 1.0, Trento Italy: Laboratory for Applied Ontology, 2003. http://wonderweb.semanticweb.org/deliverables/documents/D18.pdf.

[Meyers 01a] Meyers, B. & Oberndorf, P. *A Framework for the Specification of Acquisition Models*, (CMU/SEI-2001-TR-004, ADA401717). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. http://www.sei.cmu.edu/publications/documents/01.reports/01tr004.html.

[Meyers 01b] Meyers, B. Craig & Oberndorf, Patricia. *Managing Software Acquisition: Open Systems and COTS Products*. Boston, MA: Addison Wesley, 2001.

[Morris 04] Morris, Ed; Levine, Linda; Meyers B. Craig; Place, Patrick R. H;.& Plakosh, Daniel.
System of Systems Interoperability, (CMU/SEI-2004-TR-004) Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. http://www.sei.cmu.edu/publications/documents/04.reports/04tr004.html

[Royce 70] Royce, Winston W. "Managing the Development of Large Software Systems: Concepts and Techniques," *WESCON Technical Papers, 14*, 1-9. Los Angeles, CA: WESCON, 1970.

[Wache 01] Wache, H.; Vogele, T.; Visser, U.; Struckenschmidt. H.; Schuster, G.; Neumann, H.; & Hubner, S. "Ontology-Based integration of Information—A Survey of Existing Approaches," *IJCAI—01, Ontologies and Information Sharing*. Seattle WA, Aug.5, 2001. Breman, Germany: University of Breman, 2001.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

6. **AGENCY USE ONLY** (leave blank)	7. **REPORT DATE** March 2005	8. **REPORT TYPE AND DATES COVERED** Final
9. **TITLE AND SUBTITLE** Including Interoperability in the Acquisition Process		10. **FUNDING NUMBERS** C — F19628-95-C-0003
11. **AUTHOR(S)** B. Craig Meyers, Ira A. Monarch, Linda Levine and James D. Smith		
12. **PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		13. **PERFORMING ORGANIZATION REPORT NUMBER** CMU/SEI-2005-TR-004
14. **SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)** HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116		15. **SPONSORING/MONITORING AGENCY REPORT NUMBER** ESC-TR-2005-004
16. **SUPPLEMENTARY NOTES**		
12.a **DISTRIBUTION/AVAILABILITY STATEMENT** Unclassified/Unlimited, DTIC, NTIS		12.b **DISTRIBUTION CODE**

13. **ABSTRACT** (maximum 200 words)

This report explores achieving interoperability in the acquisition process. It asserts that interoperability applies to the management and construction of a system, as well as to its operation. This idea leads to a broader view of interoperability. Also presented is the idea that the essential character of interoperability—related to data models and operational semantics—is independent of a domain of application. This report lists a number of basic assertions that can help organizations achieve interoperability in the acquisition process. A number of related key issues are also examined. Ultimately, it is expected that achieving interoperability will depend on a formal specification of acquisition.

14. **SUBJECT TERMS** interoperability; acquisition, integration, acquisition model			15. **NUMBER OF PAGES** 57
			16. **PRICE CODE**
17. **SECURITY CLASSIFICATION OF REPORT** UNCLASSIFIED	18. **SECURITY CLASSIFICATION OF THIS PAGE** UNCLASSIFIED	19. **SECURITY CLASSIFICATION OF ABSTRACT** UNCLASSIFIED	20. **LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102